EXHIBIT A

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

PARUS HOLDINGS INC., Plaintiff, v. APPLE INC., Defendant.	C.A. No.: 6:21-cv-00968-ADA JURY TRIAL DEMANDED

PARUS HOLDING INC.'S FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Parus Holdings Inc. ("Parus" or "Plaintiff") for its First Amended Complaint for Patent infringement ("Complaint") against Apple Inc. ("Apple" or "Defendant"), hereby alleges as follows:

THE PARTIES

1. Plaintiff Parus Holdings Inc. is Delaware corporation having its principal place of business at 3000 Lakeside Drive, Suite 110S, Bannockburn, IL 60015.

2. Parus is the owner by assignment of U.S. Patent No. 6,721,705 ("the '705 Patent") (attached as Exhibit 1), U.S. Patent No. 7,516,190 ("the '190 patent") (attached as Exhibit 2), U.S. Patent No. 8,185,402 ("the '402 Patent") (attached as Exhibit 3), and U.S. Patent No. 9,769,314 ("the '314 patent") (Attached as Exhibit 4) (collectively, "the Asserted Patents").

3. Defendant Apple Inc. is a California corporation with a principal place of business at One Apple Park Way, Cupertino, California 95014.

4. Apple is registered to do business in Texas.

5. Apple has regular and established places of business in this District, including, at 3121 Palm Way, Austin, Texas, 2901 S. Capital of Texas Hwy., Austin, TX, and 12535 Riata Vista Circle, Austin, Texas, and 5501 West Parmer Lane, Austin, Texas. Apple employs thousands of people, including hundreds of engineers, who work at these locations in Texas. The work done at these Apple locations in Texas includes work related to Apple's iPhones, iPads, iPods and Mac products.

6. Apple also operates brick-and-mortar Apple Stores at Barton Creek Square, Austin, Texas and at Apple Domain Northside, Austin, Texas. Apple uses, offers for sale and sells Apple's iPhones, iPads, iPods and Mac products that include Siri functionality at these Apple Stores.

7. On information and belief, Apple can be served through its registered agent, CT Corporation System, 818 W. Seventh Street, Suite 930, Los Angeles, California, 90017.

8. Apple has placed or contributed to placing infringing products like the iPhone 12 into the stream of commerce via an established distribution channel knowing or understanding that such products would be sold and used in the United States, including in the Western District of Texas. On information and belief, Apple also has derived substantial revenues from infringing acts in the Western District of Texas, including from the sale and use of infringing products like the iPhone 12.

9. Defendant had constructive notice of the Asserted Patents based on Parus's marking at least as of 12/29/2012.

JURISDICTION AND VENUE

10. This is an action for patent infringement arising under the patent laws of the United States, Title 35 of the United States Code. Accordingly, this Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

11. This Court has specific personal jurisdiction over Defendant at least in part because Defendant conducts business in this Judicial District. Parus's causes of action arise, at least in part, from Defendant's contacts with and activities in the State of Texas and this Judicial District. Upon information and belief, each Defendant has committed acts of infringement within the State of Texas and this Judicial District by, *inter alia*, directly and/or indirectly using, selling, offering to sell, or importing products that infringe one or more claims of the '705 Patent and/or the '402 Patent.

12. Defendant has committed acts within this District giving rise to this action, and has established sufficient minimum contacts with the State of Texas such that the exercise of jurisdiction would not offend traditional notions of fair play and substantial justice.

13. Venue is proper in this Judicial District pursuant to 28 U.S.C. § 1391(b), (c), and 1400(b) because (1) Defendant has a regular and established place of business in this Judicial District, and (2) Defendant has committed and continues to commit acts of patent infringement in this Judicial District by, *inter alia*, directly and/or indirectly using, selling, offering to sell, or importing products that infringe one or more claims of the Asserted Patents.

PATENTS-IN-SUIT

14. Parus is the owner, by assignment, of the '705 Patent, titled "Robust voice browser system and voice activated device controller." A true and correct copy of the '705 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 1.

15. Parus is the owner, by assignment, of the '402 Patent, which is also titled "Robust voice browser system and voice activated device controller." A true and correct copy of the '402 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 2.

16. The '705 and '402 patents both relate to "robust and highly reliable" systems for users to search the internet using voice-enabled devices. '402 Patent at 1:15–18.[1] At the time of the invention, only a few types of devices were available for searching web sites (*i.e.*, conventional computers, PDAs, or web-phones/web-pagers). As explained in the specification, these devices had numerous limitations, including (i) the form of the devices, their locations, and their ability to connect to the Internet; (ii) the limited compatibility of the devices with particular web site designs; and (iii) the devices' limited responsiveness to rapid changes in website content (e.g., "[t]he design of the web site may change, the information required by the web site in order to perform searches may change, and the method of reporting search results may change"). *Id*. at 2:25–36. Therefore, there was a need for a system that could "detect modifications to web sites and adapt to such changes in order to quickly and accurately provide the information requested by a user through a voice enabled device." *Id*. at 2:32–36.

17. Voice-enabled searches of the Internet present several unique technological hurdles. For example, unlike regular browser-based or application-based searches, a voice-enabled device must limit its results because a user simply cannot listen to an entire page worth of search results. *See id.* at 2:37–52. Voice users are especially sensitive to latency and expect immediate responses to their search requests. *Id.* at 2:40-42. Indeed, rapid responses are an essential feature of a voice system's desirability and usability. *Id*. at 2:44–47. And "[a] system that introduces too much delay between the time a user makes a request and the time of response will not be tolerated by users and will lose its usefulness." *Id*.

18. The inventors of the '705 and '402 patents were thus presented with a technical problem: how to quickly provide complete, timely, and relevant web site search results to voice-

[1] The '705 and '402 patents share a common specification.

enabled devices, accounting for the rapidly changing nature of web sites and Internet applications. *Id.* at 1:61-2:11, 2:20–52.

19. To address this need in the art, the inventors of the Patents-in-Suit developed specific and concrete ways of solving the technical problems presented by voice-based internet searching: systems and methods for sequentially accessing web sites based on a ranked order and periodically polling web sites on the Internet. The Patents-in-Suit are directed to a "robust" system to provide quick, reliable results to the voice-based user that can access web sites in a ranked order in response to a voice request and "on its own" discover new web sites by searching for new sources on the Internet. *See, e.g.,* '705 Patent at 2:64-3:4; 3:9-12; 3:17-22.

20. When viewed as a whole, the claims of the '705 and '402 patents contain elements, including when viewed as an ordered combination, that are unconventional and were not routine or not merely a recitation of well-understood technologies or components at the time of the invention. This fact is underscored by the specification's clear explanation of the state of the art and of the need for the inventors' technological improvement to voice-based internet searching. The claims recite a specific, discrete implementation of a method and system for voice-enabled searching of web sites. The claimed inventions were not well-known, routine, or conventional technologies or components at the time of the invention and represent specific improvements over the prior art and existing systems and methods. The claimed technology was not known in the prior art at the time of the invention, let alone well-known, routine, or conventional.

21. Each of the '705 and '402 patents have separate claims. Claims are defined by their language, and the claims of each Patent-in-Suit vary in scope. For example, claims 1 and 2 of the '705 Patent relate to an internet browsing system and method, while claims 3 and 4 of the

'705 Patent relate to a system and method for controlling household devices. Likewise, claim 6 of the '402 Patent requires the method to periodically poll each web site. No one claim is representative of every claim in either patent. Although each claim recites a series of elements that, when taken in combination are not conventional, well-understood, or routine, because the elements of each claim are vary, the analysis of conventionality will vary as well.

22. The inventors of the '190 and '314 patents were likewise presented with a technological problem: how to best serve remote users with personalized networked information via an audio device, such as a telephone — without the need for a computer. Thus, the inventors of the '190 and '314 patents created methods and systems for secure, reliable retrieval of information over the Internet, where the information could be delivered over an audio device, using voice commands that could be personalized for each user. *Id.* at 1:51–2:5. This improvement in technology employed specific technological components and processes, including, *inter alia*, user-generated descriptor files, content extraction, and clipping clients. *See id.* at 2:6–4:51. The shared specification even offers detailed, exemplary source code, reinforcing the technological nature of the invention.

COUNT I

APPLE'S INFRINGEMENT OF U.S. PATENT NO. 6,721,705

23. Parus restates and incorporates by reference all of the allegations made in the preceding paragraphs as though fully set forth herein.

24. Parus is the owner, by assignment, of the '705 Patent. A true copy of the '705 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 1.

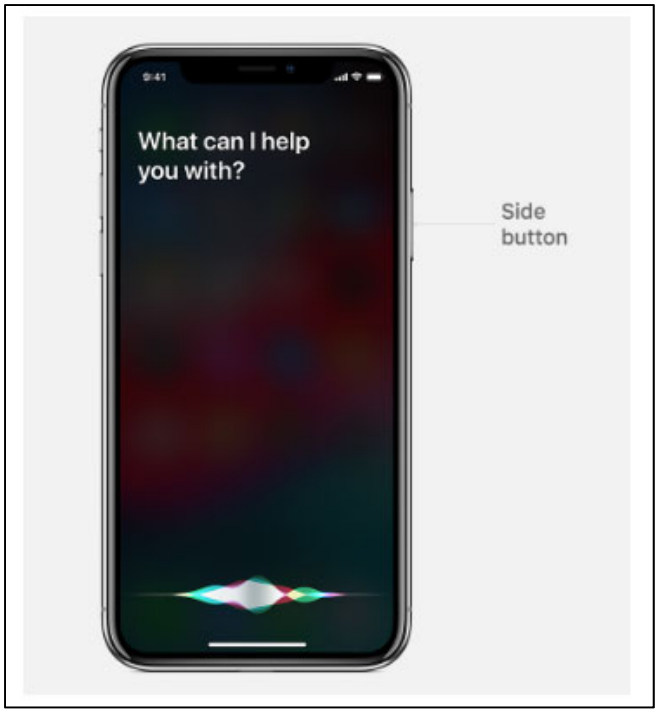
25. Defendant Apple has directly infringed, and is continuing to directly infringe, literally or under the doctrine of equivalents, at least independent claim 2 of Parus's '705 Patent

by making, using, selling, and/or offering for sale its Apple devices with Siri in the United States, in violation of 35 U.S.C. § 271(a).

26. At least as of the filing of this complaint, Defendant Apple has knowledge of the '705 Patent.

27. Various products with Siri made or sold by Apple directly infringe at least independent claim 2 of the '705 Patent. Those Apple products include at least the Apple iPhone 6s or later models, iPad Pro 12.9 inch (3rd Generation), iPad Pro 11-inch, iPad Pro 12.9-inch (2nd Generation), iPad Pro 10.5 inch, iPad Pro 9.7 inch, iPad (6th Generation), all Apple iWatches, all HomePods, CarPlay, MacBook Pro (15 inch, 2018), MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports), MacBook Air (Retina, 13-inch, 2018), and iMac Pro. (Apple Accused Products). *See e.g.*, https://support.apple.com/en-us/HT209014.

28. Each of the Apple Accused Products in conjunction with Siri perform a method for using voice commands to browse Internet web sites as required by claim 2 of the '705 Patent. As a way of illustration, the Apple iPhone X with Siri is a voice enabled device that allows users to utter speech commands into a voice enabled device and provide users with retrieved information from pre-selected web sites in an audio form via said voice enabled device:

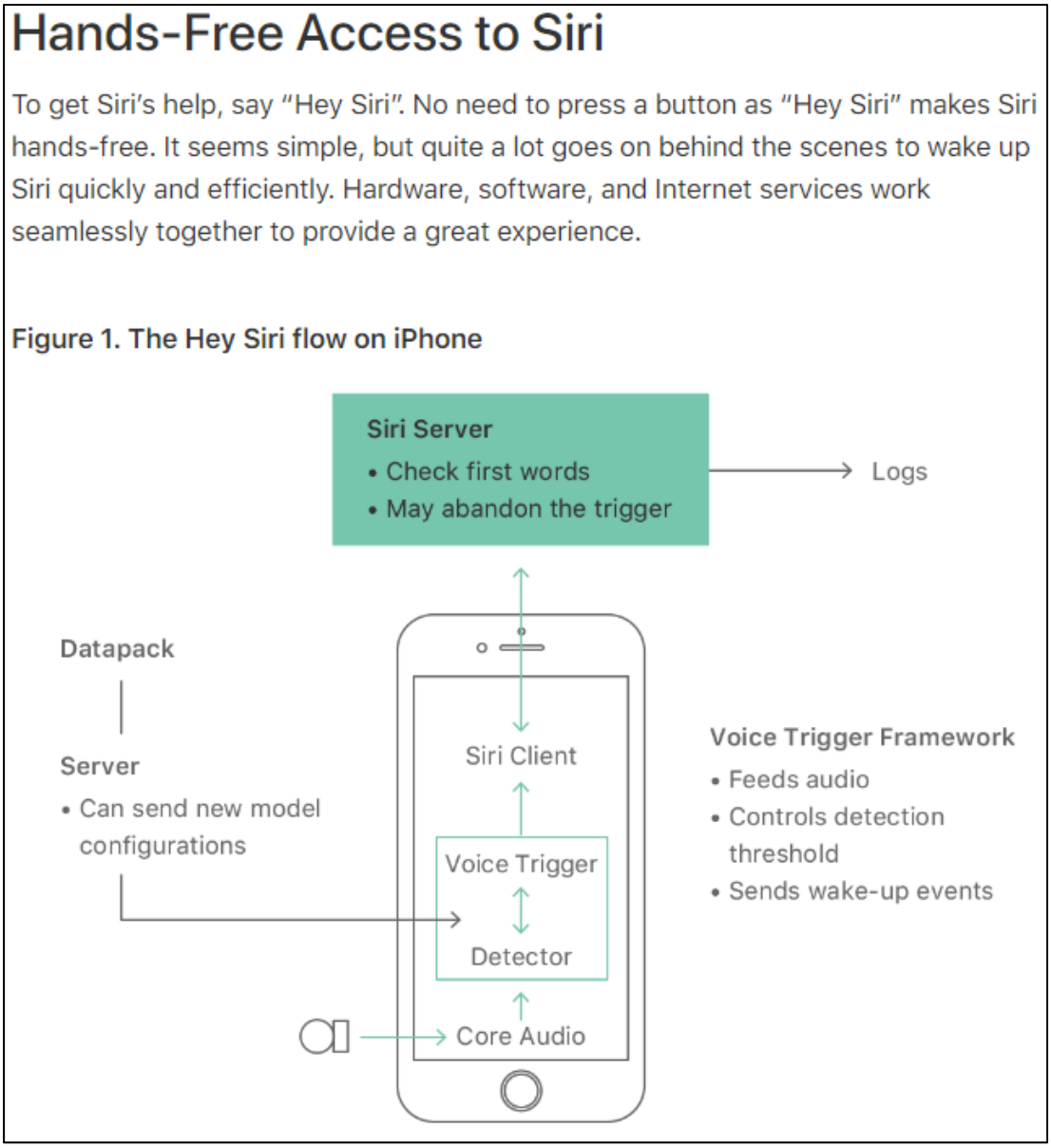


See e.g., https://support.apple.com/en-us/HT204389. Based on information and belief, those preselected websites are provided in a database as required by claim 2 of the '705 Patent.

29. Siri allows the user to communicate with the iPhone X using voice recognition and speech synthesis.

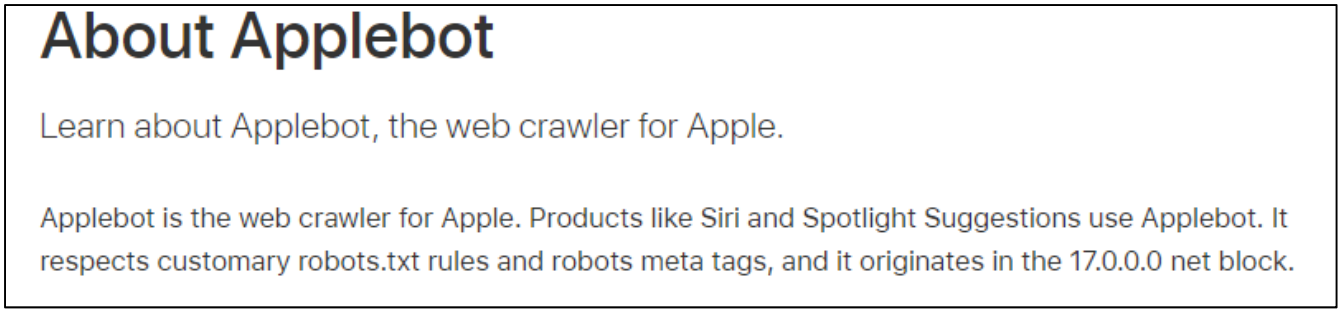
> Siri is a personal assistant that communicates using speech synthesis. Starting in iOS 10 and continuing with new features in iOS 11, we base Siri voices on deep learning. The resulting voices are more natural, smoother, and allow Siri's personality to shine through. This article presents more details about the deep learning based technology behind Siri's voice.

See e.g., https://machinelearning.apple.com/2017/08/06/siri-voices.html.



See e.g., https://machinelearning.apple.com/2017/10/01/hey-siri.html.

30. The Apple iPhone X in conjunction with Siri acquires information from and/or via one or more sources providing a database storing a list of websites on disk or in memory and assigning a rank number to each of the web sites and storing the rank number in the database. For example, the Apple iPhone X in conjunction with Siri uses a list of web sites that have been already crawled to obtain information.



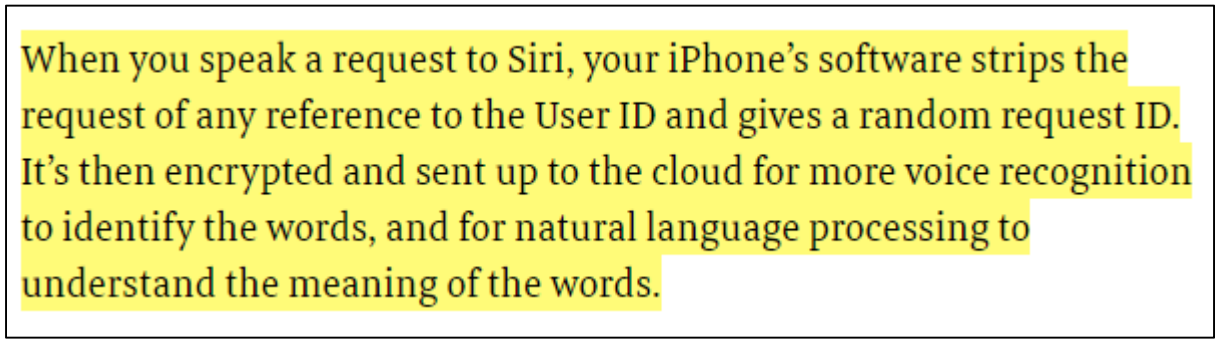
About Applebot

Learn about Applebot, the web crawler for Apple.

Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

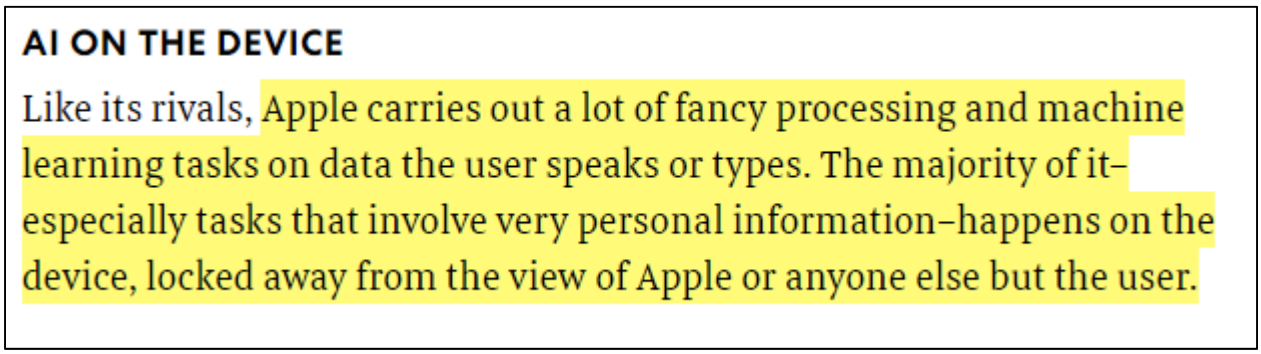
31. Further, the Apple iPhone X uses the cloud to receive a voice command from a user and converting the command into a digital data message by performing natural language processing to understand the user's spoken commands and convert the command into a digital data message.



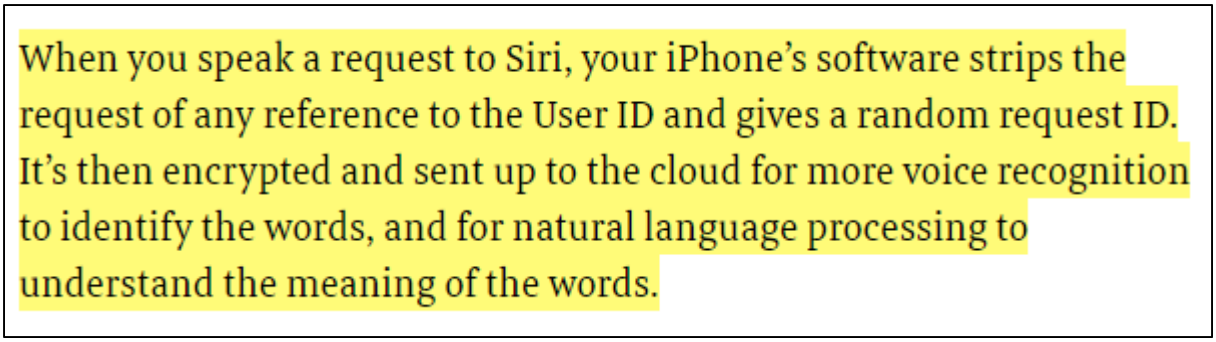
When you speak a request to Siri, your iPhone's software strips the request of any reference to the User ID and gives a random request ID. It's then encrypted and sent up to the cloud for more voice recognition to identify the words, and for natural language processing to understand the meaning of the words.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

32. The Apple Accused Products in conjunction with Siri includes a CPU based web browsing system for receiving the digital data message and accessing one of the web sites having the highest rank number. For example, the Apple iPhone X includes a CPU that identifies the information to be retrieved. Because the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or with collaboration with the cloud, there is a CPU-based web browsing system for receiving the digital data message and accessing one of the web sites having the highest ranking as required by claim 2 of the '705 Patent.



See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.



See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy

33. The Apple Accused Products in conjunction with Siri provide a database storing a list of web sites on a disk or memory. For example, the Apple iPhone X includes a plurality of pre-selected web site addresses, and each pre-selected web site address identifying a web site where the information may be retrieved. These websites have previously been crawled and pre-selected by Applebot:

> Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.
>
> > *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.

> **About Applebot**
>
> Learn about Applebot, the web crawler for Apple.
>
> Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

34. Further, the Apple iPhone X in conjunction with Siri uses Google and Bing to assist with providing a plurality of pre-selected web site addresses, each said web site address identifying a web site containing said information to be retrieved. Those web sites are identified in a ranked order in the database.

> The big winner in Apple's recent switch to Google for search may be Siri. Last week Apple confirmed that its personal assistant will still use Microsoft's Bing for its image searches, but that Google will provide the web search, and YouTube will provide the video search. Apple will also use Google for searches in iOS, and in Spotlight in macOS.

See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.

> "Google is much better at extracting information from web content and presenting it to the user in a list format, which is created on a query by query basis," says Scott Zimmerman, technical SEO analyst at Walker Sands. "The quality of results, especially as it relates to long-tail queries that are typical for voice search, are significantly better on Google."

See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.

35. The Apple iPhone X in conjunction with Siri is a voice-enabled device that includes a CPU-based web browsing system for receiving the digital data message and accessing one of the web sites having the highest rank number, the web browsing system including at least a content extraction agent, a content fetcher, a polling and ranking agent and a content file. The system provides information retrieved from the web sites to the user in an audio form via the voice-enabled device.

> **What can Siri do?**
>
> Siri is the digital assistant that's built into your iPhone, iPad, Apple TV, Apple Watch and Mac. You can ask Siri to do something and have your device do it, rather than go through multiple menus or tap through lots of options.
>
> Siri is constantly improving to become faster and more reliable, and to expand its repertoire of functions. In iOS 7.1 it gained a female voice for the UK, iOS 8 added music identification and "'Hey Siri!", and iOS 9 made Siri contextually aware. iOS 10 gave it the ability to control non-Apple apps, and this coincided with it arriving on the Mac. In iOS 11 learned to translate phrases and iOS 12 introduced Siri shortcuts.

See e.g., https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

36. The Apple Products in conjunction with Siri include a computer, said computer operatively connected to the internet.

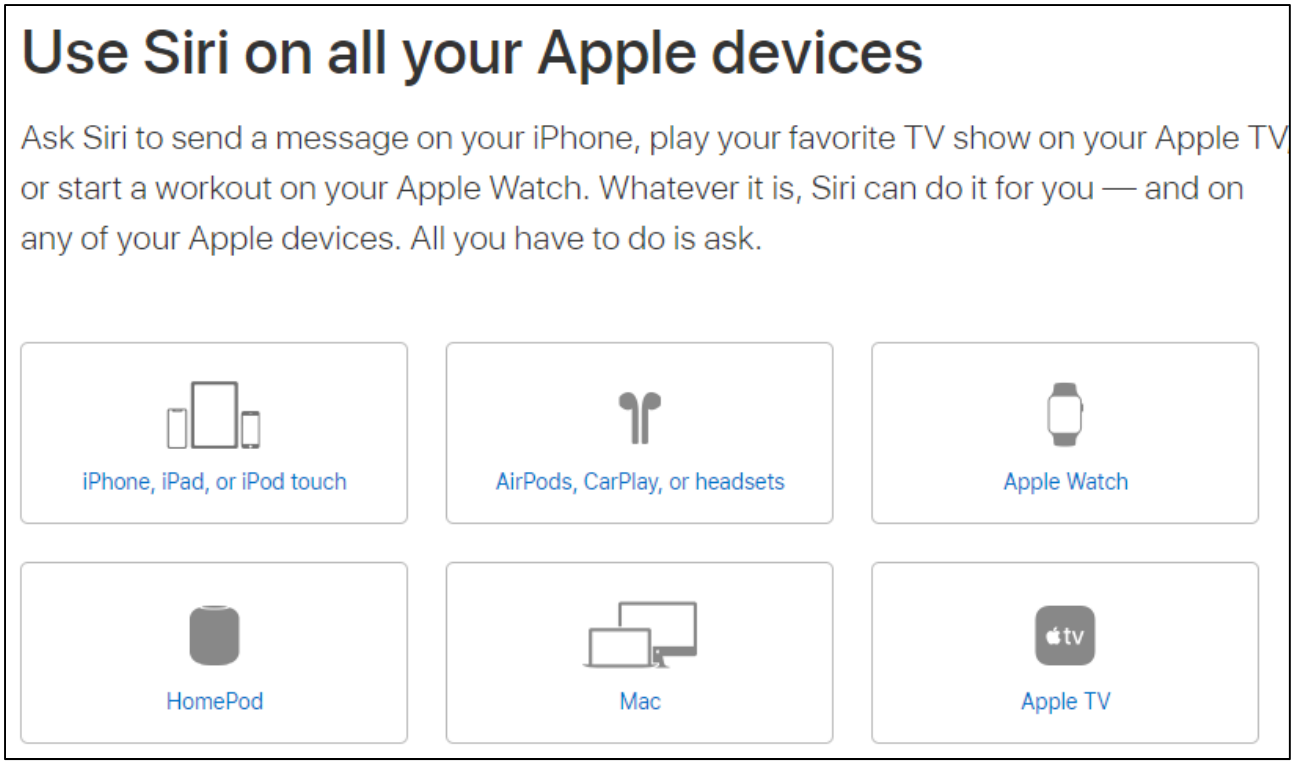
37. For example, the Apple iPhone X has a computer (i.e., microprocessor) in the A11 integrated circuit. https://support.apple.com/kb/sp770?locale=en_US. The A11 integrated circuit is operatively coupled to the internet. *Id.* As an additional example, and alternatively, the Apple iPhone X with built-in Siri has access to computing hardware that processes questions asked of Siri including 32 powerful HP servers with a total of 1024 cores and 32 terabytes of RAM a piece[2]. Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM. The company also says its text-to-speech can run on *"both general and* special purpose microprocessors, and any one or more processors of any kind of digital computer," indicating high sophistication and optimization.

> This information allegedly comes — albeit second-hand — from Apple's lead cloud architect, who says that every instance of Siri runs on 32 powerful HP servers with a total of 1024 cores and 32 terrabytes of RAM apiece. That certainly makes the new Mac Pro look long in the tooth.
>
> Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM.

See e.g., https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.

38. Further, the computer is operatively connected to the internet and operatively connected to the cloud:

[2] https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.



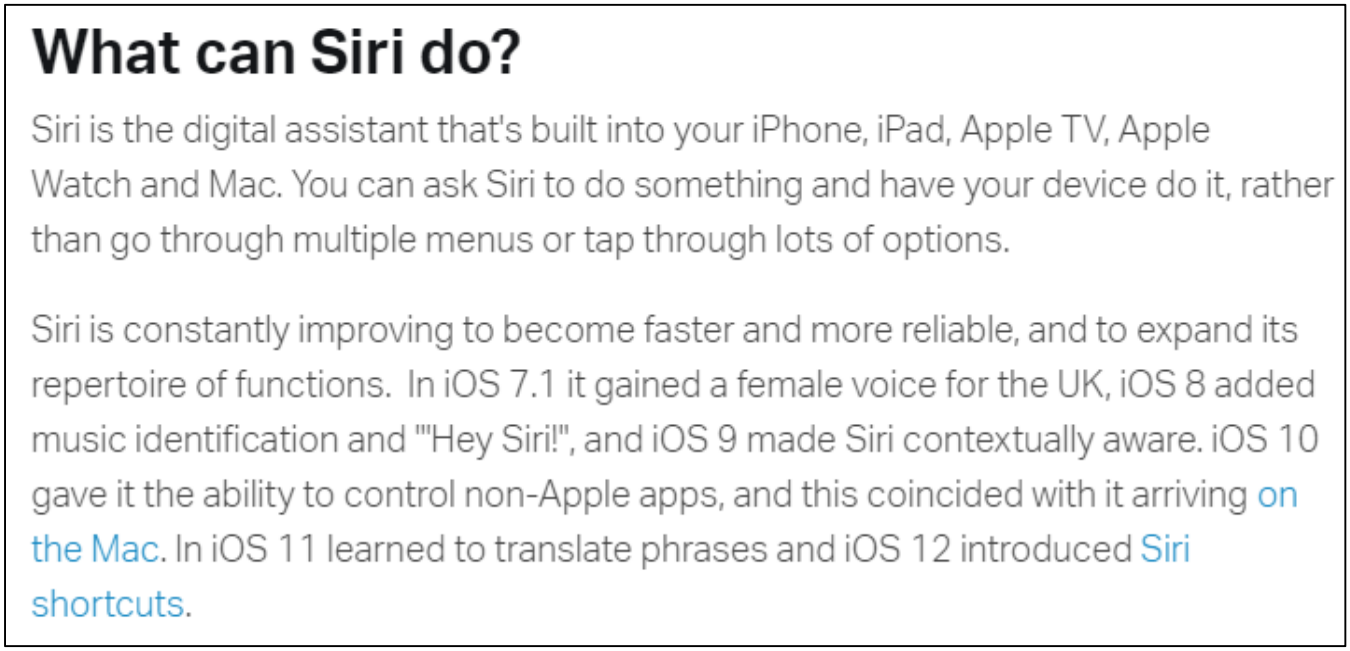
See e.g., https://support.apple.com/en-us/HT204389.

39. The Apple Accused Products in conjunction with Siri serve as a voice enabled device operatively connected to said computer, said voice enabled device configured to receive speech commands from users. For example, the Apple iPhone X with built-in Siri (controlled by AI models in the cloud) includes a voice enabled device operatively connected to the computing hardware, the voice enabled device (the microphone on the Apple iPhone X and associated processing power) is configured to receive speech commands from users. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

40. The Apple Accused Products in conjunction with Siri include at least one speaker-independent speech recognition device, said speaker-independent speech recognition device operatively connected to said computer and to said voice enabled device. For example, the Apple iPhone X with built-in Siri (controlled by AI models in the cloud) includes at least one speaker-independent speech recognition device, said speaker-independent speech recognition

device operatively connected to the computing hardware and to the voice enabled Apple iPhone X. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

41. The Apple Accused Products in conjunction with Siri receive web browsing system response data from the web site with the highest rank number and convert the response data into an audio message that is transmitted to the user. For example, the Apple iPhone X in conjunction with Siri is a system for retrieving information from web sites by uttering speech commands into a voice enabled device. Therefore, the speech commands comprise information requests selectable by the user.



What can Siri do?

Siri is the digital assistant that's built into your iPhone, iPad, Apple TV, Apple Watch and Mac. You can ask Siri to do something and have your device do it, rather than go through multiple menus or tap through lots of options.

Siri is constantly improving to become faster and more reliable, and to expand its repertoire of functions. In iOS 7.1 it gained a female voice for the UK, iOS 8 added music identification and "'Hey Siri!", and iOS 9 made Siri contextually aware. iOS 10 gave it the ability to control non-Apple apps, and this coincided with it arriving on the Mac. In iOS 11 learned to translate phrases and iOS 12 introduced Siri shortcuts.

See e.g., https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

42. The Apple Accused Products in conjunction with Siri include said computer further configured to access at least one of said plurality of web sites identified by said instruction set to obtain said information to be retrieved, aid computer configured to first access the web site with the highest rank number. For example, the Apple iPhone X with built-in Siri is a system for retrieving information from pre-selected web sites by uttering speech commands

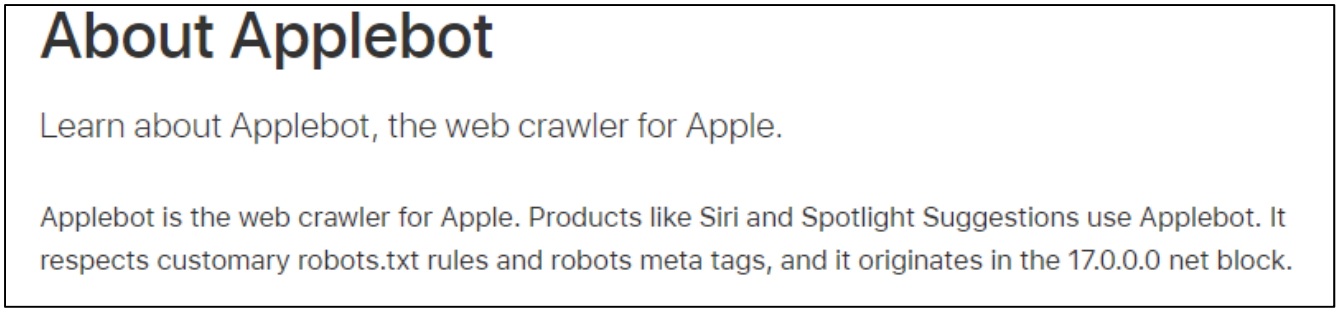
into a voice enabled device. Apple references Siri on its website as the virtual assistant that recognizes voice commands and executes the requested functions. Siri uses various sources for obtaining information, including search engines, websites, or apps on a device. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

43. The Apple Accused Products in conjunction with Siri include at least one speech synthesis device, said speech synthesis device operatively connected to said computer and to said voice enabled device for converting response data from a website into an audio message that is transmitted to a user. For example, the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or with help from the cloud. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

44. Further, the Apple iPhone X in conjunction with Siri uses Applebot to crawl a plurality of web sites.

> Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.
>
> > *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.



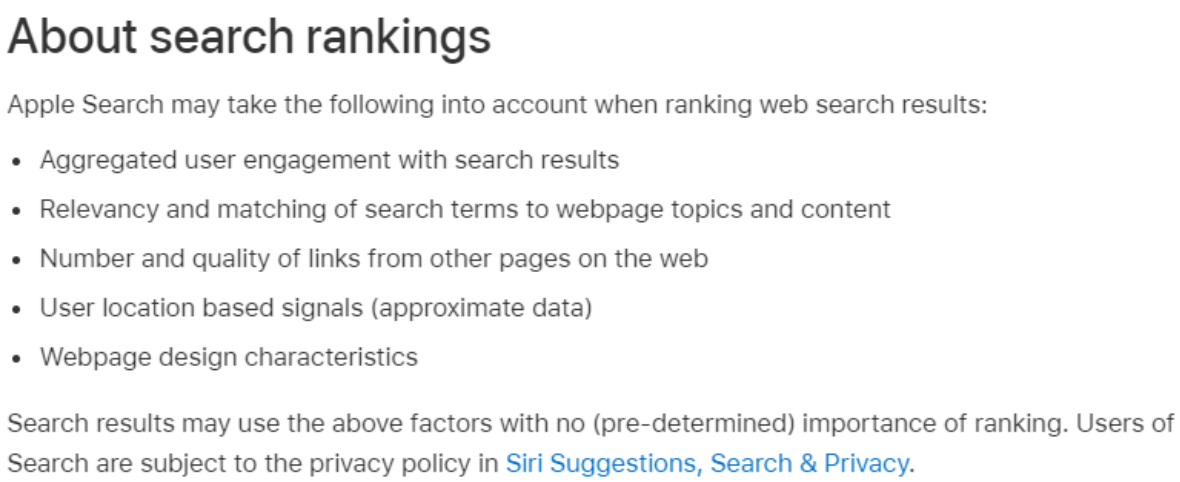
About Applebot

Learn about Applebot, the web crawler for Apple.

Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

45. The Apple iPhone X in conjunction with Siri uses Applebot to crawl a plurality of web sites, periodically polls each of the web sites, decreases the rank number of the polled web site if no response is received, if an unexpected response is received, and if a response time of the polled web site is longer than a second response time of a second polled web site.



About search rankings

Apple Search may take the following into account when ranking web search results:

- Aggregated user engagement with search results
- Relevancy and matching of search terms to webpage topics and content
- Number and quality of links from other pages on the web
- User location based signals (approximate data)
- Webpage design characteristics

Search results may use the above factors with no (pre-determined) importance of ranking. Users of Search are subject to the privacy policy in Siri Suggestions, Search & Privacy.

See e.g., https://support.apple.com/en-us/HT204683.

46. In addition to Applebot, the Apple iPhone X in conjunction with Siri uses Google and Bing for searches as well. *See e.g.*, https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search. For example, the Google search engine uses a wide variety of polling mechanisms to determine the quality of a webpage and to change the rank of the site, including using polling digital data message and whether a response is received from a polled web site. *See, e.g.*,

https://developers.google.com/search/docs/advanced/guidelines/cloaking; https://stackoverflow.com/questions/1878364/how-does-google-know-you-are-cloaking; https://www.google.com/search/howsearchworks/algorithms/ (Quality of content); https://developers.google.com/search/docs/advanced/guidelines/webmaster-guidelines. Further, Microsoft's Bing uses a wide variety of polling mechanisms to determine the quality of a webpage and to change the rank of the site, including using polling digital data message and whether a response is received from a polled web site. *See, e.g.*, https://www.bing.com/webmasters/help/webmaster-guidelines-30fba23a.

47. The Apple Accused Products in conjunction with Siri include said speech synthesis device configured to produce an audio message containing any retrieved information from said pre-selected web sites; and said speech synthesis device further configured to transmit said audio message to said users via said voice enabled device. For example, the Apple iPhone X includes the speech synthesis device configured to produce an audio message containing any retrieved information from the pre-selected web sites, and the speech synthesis device further configured to transmit said audio message to said users via said voice enabled device. Because the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or in collaboration with the cloud, there is a recognition grammar corresponding to each instruction set and corresponding speech command. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

48. Defendant Apple has also infringed, and continues to infringe, claims of the '705 Patent by offering to commercially distribute, commercially distributing, selling, making and/or

importing the Apple Accused Products, which are used in practicing the process, or using the systems, of the '705 Patent, and constitute a material part of the invention.

COUNT II

APPLE'S INFRINGEMENT OF U.S. PATENT NO. 8,185,402

49. Parus restates and incorporates by reference all of the allegations made in the preceding paragraphs as though fully set forth herein.

50. Parus is the owner, by assignment, of the '402 Patent. A true copy of the '402 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 2.

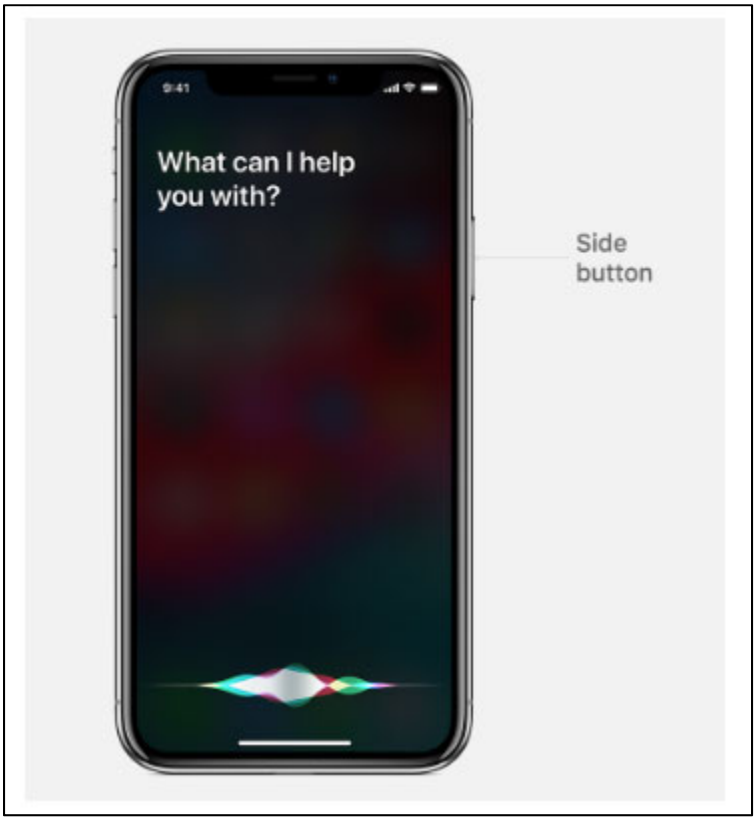
51. Defendant Apple has directly infringed, and is continuing to directly infringe, literally or under the doctrine of equivalents, at least independent claim 1 of Parus's '402 Patent by making, using, selling, and/or offering for sale its Apple devices with Siri in the United States, in violation of 35 U.S.C. § 271(a).

52. At least as of the filing of the original complaint, Defendant Apple has knowledge of the '402 Patent.

53. Various products with Siri made or sold by Apple directly infringe at least independent claim 1 of the '402 Patent. Those Apple products include at least the Apple iPhone X or later models, the Apple iPhone 6s or later models, iPad Pro 12.9 inch (3rd Generation), iPad Pro 11-inch, iPad Pro 12.9-inch (2nd Generation), iPad Pro 10.5 inch, iPad Pro 9.7 inch, iPad (6th Generation), all Apple iWatches, all HomePods, CarPlay, MacBook Pro (15 inch, 2018), MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports), MacBook Air (Retina, 13-inch, 2018), and iMac Pro. (Apple Accused Products). *See e.g.*, https://support.apple.com/en-us/HT209014.

54. The Apple Accused Products in conjunction with Siri perform a method for retrieving information from web sites by uttering speech commands into a voice enabled device

and for providing to users retrieved information in an audible form via said voice enabled device. For example, the Apple iPhone X in conjunction with Siri is a voice enabled device:



See e.g., https://support.apple.com/en-us/HT204389; *see also*, https://machinelearning.apple.com/2017/08/06/siri-voices.html; https://machinelearning.apple.com/2017/10/01/hey-siri.html.

55. The Apple Accused Products in conjunction with Siri includes at least one computing device, the computing device operatively coupled to one or more networks. For example, the Apple iPhone X has a computer (i.e., microprocessor) in the A11 integrated circuit. https://support.apple.com/kb/sp770?locale=en_US. The A11 integrated circuit is operatively coupled to the internet. *Id.*

56. As an additional example, and alternatively, the Apple iPhone X with built-in Siri has access to computing hardware that processes questions asked of Siri including 32 powerful

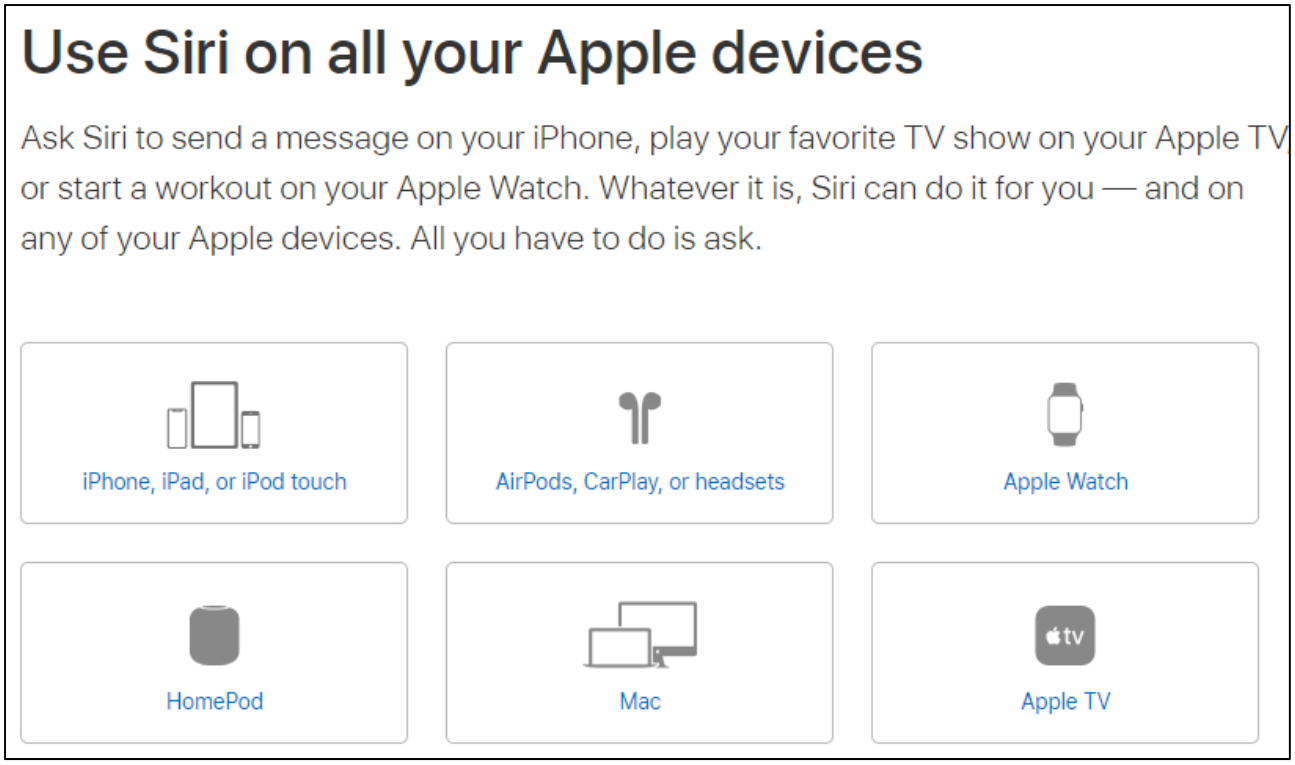
HP servers with a total of 1024 cores and 32 terabytes of RAM a piece[3]. Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM. The company also says its text-to-speech can run on *"both general and* special purpose microprocessors, and any one or more processors of any kind of digital computer," indicating high sophistication and optimization.

> This information allegedly comes — albeit second-hand — from Apple's lead cloud architect, who says that every instance of Siri runs on 32 powerful HP servers with a total of 1024 cores and 32 terrabytes of RAM apiece. That certainly makes the new Mac Pro look long in the tooth.
>
> Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM.

See e.g., https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.

57. Further, the Apple iPhone X in conjunction with Siri provides a computer operatively coupled to the internet.

[3] https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.



See e.g., https://support.apple.com/en-us/HT204389.

58. The Apple Accused Products in conjunction with Siri also provide the computer further being operatively connected to at least one speaker-independent speech recognition engine and to at least one speech synthesis engine. For example, the Apple iPhone X in conjunction with Siri is a speaker-independent speech-recognition device, the speaker-independent speech-recognition device operatively connected to the computing device and configured to receive the speech commands. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://machinelearning.apple.com/2017/10/01/hey-siri.html.

59. The Apple iPhone X in conjunction with Siri provides a voice enabled device operatively connected to the computer and is configured to receive speech commands from users.

How Siri Works

Upon receiving your request, Siri records the frequencies and sound waves from your voice and translates them into a code. Siri then breaks down the code to identify particular patterns, phrases, and keywords. This data gets input into an algorithm that sifts through thousands of combinations of sentences to determine what the inputted phrase means. This algorithm is complex enough that it is capable of working around idioms, homophones and other literary expressions to determine the context of a sentence.

Once Siri determines its request, it begins to assess what tasks needs to be carried out, determining whether or not the information needed can be accessed from within the phone's data banks or from online servers. Siri is then able to craft complete and cohesive sentences relevant to the type of question or command requested.

See e.g., https://www.jameco.com/Jameco/workshop/howitworks/how-siri-works.html.

When you speak a request to Siri, your iPhone's software strips the request of any reference to the User ID and gives a random request ID. It's then encrypted and sent up to the cloud for more voice recognition to identify the words, and for natural language processing to understand the meaning of the words.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

AI ON THE DEVICE

Like its rivals, Apple carries out a lot of fancy processing and machine learning tasks on data the user speaks or types. The majority of it–especially tasks that involve very personal information–happens on the device, locked away from the view of Apple or anyone else but the user.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy; *see also* https://machinelearning.apple.com/2017/10/01/hey-siri.html.

60. The Apple Accused Products in conjunction with Siri provide a speech command to the speaker-independent speech recognition engine. Because the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or with help from the cloud, the Apple Accused Products in conjunction with Siri provide the speech command to the speaker-independent speech recognition engine. *See e.g.,* https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

AI ON THE DEVICE

Like its rivals, Apple carries out a lot of fancy processing and machine learning tasks on data the user speaks or types. The majority of it–especially tasks that involve very personal information–happens on the device, locked away from the view of Apple or anyone else but the user.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

61. The Apple Accused Products in conjunction with Siri have the computer accessing at least one of a plurality of web sites associated with the speech command to obtain information to be retrieved , the computer first accessing a first web site of the plurality of web sites and, if the information to be retrieved is not found at the first web site, the computer sequentially accessing the plurality of web sites until the information to be retrieved is found or until the plurality of web sites has been accessed. For example, the Apple iPhone X in conjunction with Siri includes a plurality of web site addresses, each web site address identifying a web site containing the information to be retrieved. These websites have previously been crawled by Applebot:

Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.

> *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.

About Applebot

Learn about Applebot, the web crawler for Apple.

Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

62. Further, the Apple iPhoneX in conjunction with Siri uses Google and Bing to assist with providing a plurality of web site addresses, each said web site address identifying a web site containing said information to be retrieved.

The big winner in Apple's recent switch to Google for search may be Siri. Last week Apple confirmed that its personal assistant will still use Microsoft's Bing for its image searches, but that Google will provide the web search, and YouTube will provide the video search. Apple will also use Google for searches in iOS, and in Spotlight in macOS.

See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.

> "Google is much better at extracting information from web content and presenting it to the user in a list format, which is created on a query by query basis," says Scott Zimmerman, technical SEO analyst at Walker Sands. "The quality of results, especially as it relates to long-tail queries that are typical for voice search, are significantly better on Google."

See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.

63. For example, because the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or with help from the cloud, the device itself may recognize a speech command.

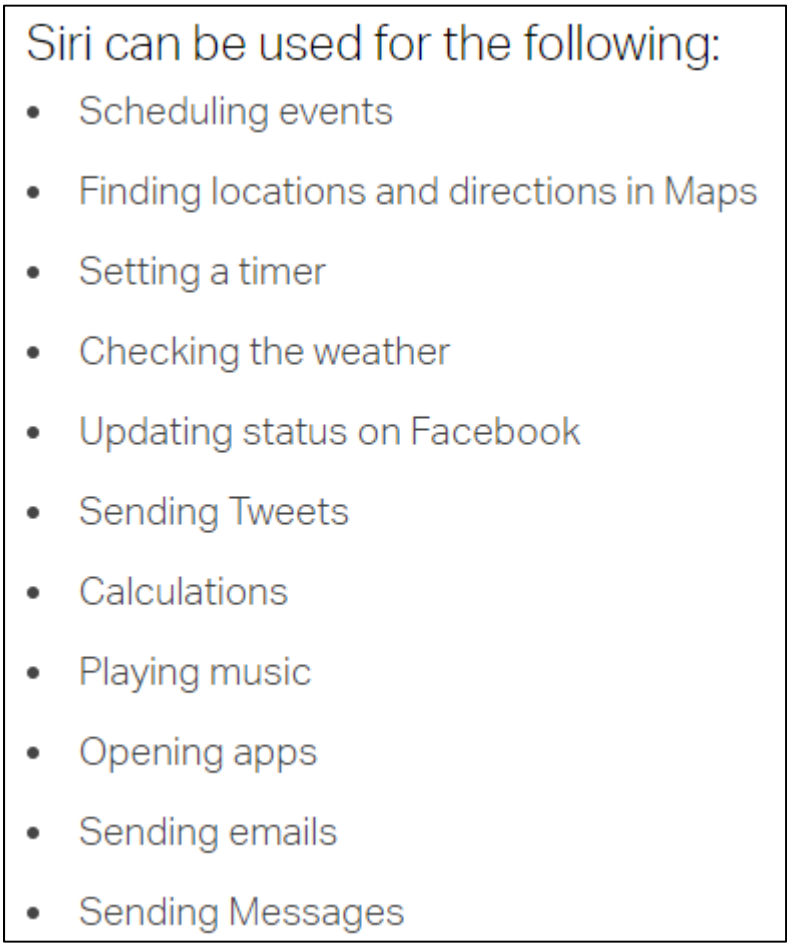
> **AI ON THE DEVICE**
>
> Like its rivals, Apple carries out a lot of fancy processing and machine learning tasks on data the user speaks or types. The majority of it–especially tasks that involve very personal information–happens on the device, locked away from the view of Apple or anyone else but the user.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

> When you speak a request to Siri, your iPhone's software strips the request of any reference to the User ID and gives a random request ID. It's then encrypted and sent up to the cloud for more voice recognition to identify the words, and for natural language processing to understand the meaning of the words.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

64. Further, the Apple iPhone X in conjunction with Siri is a system for retrieving information from web sites by uttering speech commands into a voice enabled device. Therefore, the speech commands comprise information requests selectable by the user. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.



See e.g., https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

65. Also, the Apple iPhone X in conjunction with Siri includes the speaker-independent speech-recognition device configured to receive the speech command from the users via the voice-enabled device and to select the corresponding recognition grammar upon receiving

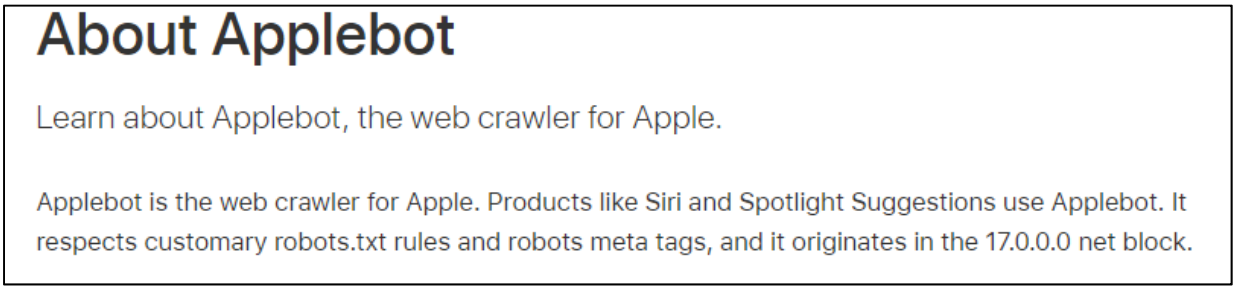
the speech command. *See e.g.,* https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

66. The Apple Accused Devices in conjunction with Siri includes the computer first accessing a first web site of the plurality of web sites and, if the information to be retrieved is not found at the first web site, the computer sequentially accessing the plurality of web sites until the information to be retrieved is found or until the plurality of web sites has been accessed. For example, the Apple iPhone X with built-in Siri is a system for retrieving information from pre-selected web sites by uttering speech commands into a voice enabled device. Apple references Siri on its website as the virtual assistant that recognizes voice commands and executes the requested functions. Siri uses various sources for obtaining information, either by delegating searches to search engines (Bing or Google) or using websites such as Yelp and others, or by accessing any one of several apps that it has on its device. *See e.g.,* https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

67. Further, the Apple iPhone X in conjunction with Siri use Applebot to crawl a plurality of web sites.

> Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.
>
> > *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.



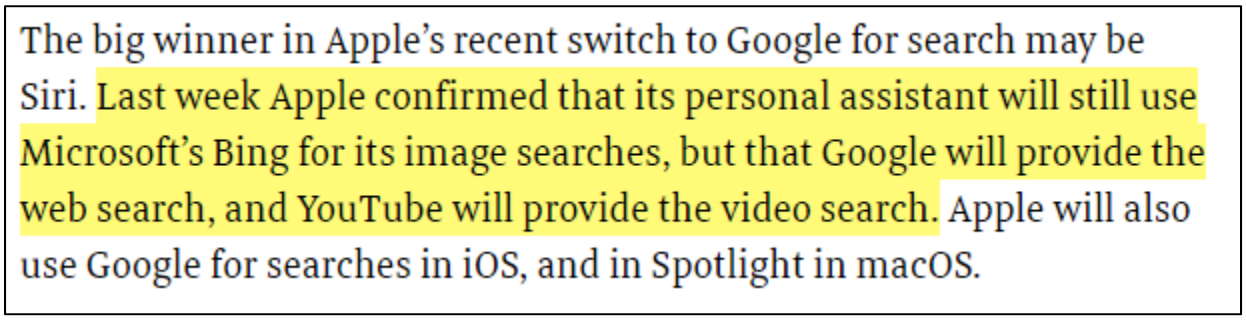
About Applebot

Learn about Applebot, the web crawler for Apple.

Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

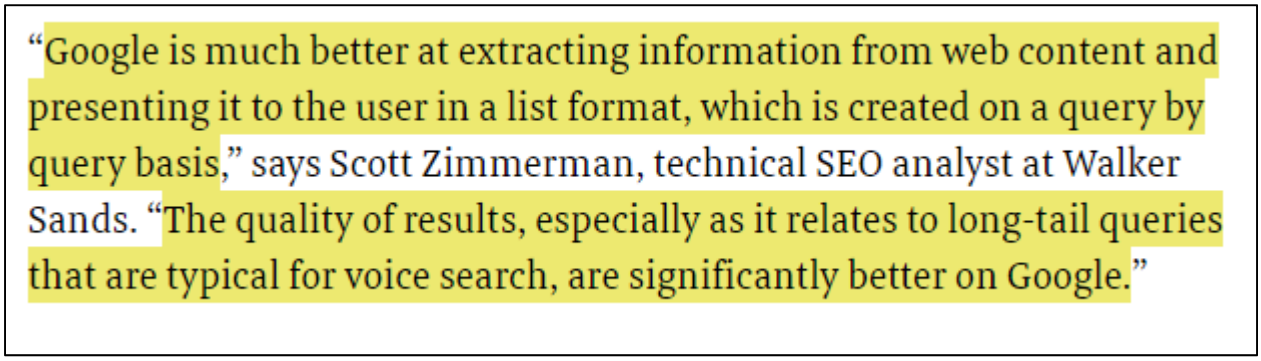
See e.g., https://support.apple.com/en-us/HT204683.

68. In addition to Applebot, the Apple iPhone X in conjunction with Siri use Google and Bing for search as well:



The big winner in Apple's recent switch to Google for search may be Siri. Last week Apple confirmed that its personal assistant will still use Microsoft's Bing for its image searches, but that Google will provide the web search, and YouTube will provide the video search. Apple will also use Google for searches in iOS, and in Spotlight in macOS.

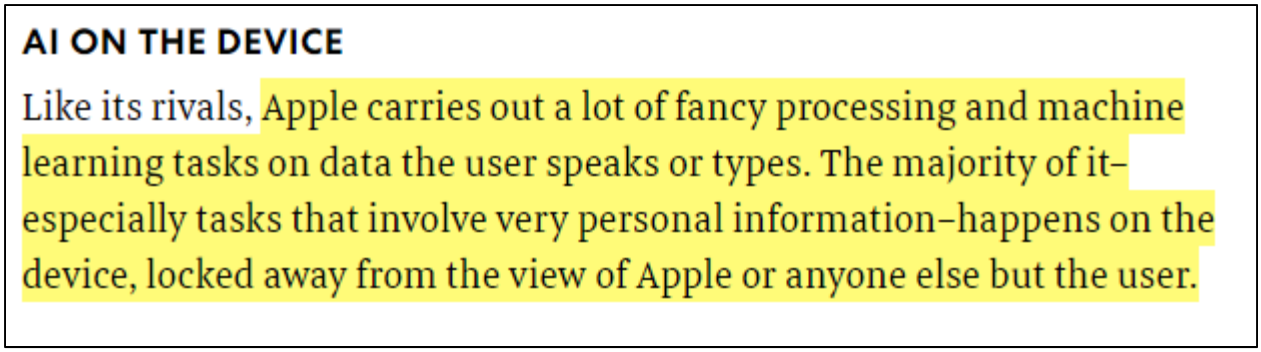
See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.



"Google is much better at extracting information from web content and presenting it to the user in a list format, which is created on a query by query basis," says Scott Zimmerman, technical SEO analyst at Walker Sands. "The quality of results, especially as it relates to long-tail queries that are typical for voice search, are significantly better on Google."

See e.g., https://www.fastcompany.com/40475434/siri-may-be-the-big-winner-in-apples-switch-to-google-for-web-search.

69. The Apple Accused Products in conjunction with Siri include the speech synthesis engine producing an audio message containing any retrieved information from the web sites and transmitting the audio message to the users via the voice enabled device. For example, the Apple iPhone X in conjunction with Siri includes the speech synthesis device configured to produce an audio message containing any retrieved information from the plurality of web sites, and the speech synthesis device further configured to transmit said audio message to said users via said voice enabled device. Because the Apple iPhone X in conjunction with Siri can receive and process voice commands on the device itself or in collaboration with the cloud, there is a recognition grammar corresponding to each instruction set and corresponding speech command. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.



AI ON THE DEVICE

Like its rivals, Apple carries out a lot of fancy processing and machine learning tasks on data the user speaks or types. The majority of it–especially tasks that involve very personal information–happens on the device, locked away from the view of Apple or anyone else but the user.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

70. The Apple Accused Products in conjunction with Siri include the speech synthesis device further configured to transmit the audio message to the users via the voice-enabled device. For example, the Apple iPhone X in conjunction with Siri transmits the audio message to the user via the voice enabled device. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

71. Defendant Apple has had knowledge of the '402 Patent since at least the filing of the complaint.

72. Defendant Apple has also infringed, and continues to infringe, claims of the '705 Patent by offering to commercially distribute, commercially distributing, selling, making and/or importing the Apple Accused Products, which are used in practicing the process, or using the systems, of the '705 Patent, and constitute a material part of the invention.

COUNT III

APPLE'S INFRINGEMENT OF U.S. PATENT NO. 7,516,190

73. Parus restates and incorporates by reference all of the allegations made in the preceding paragraphs as though fully set forth herein.

74. Parus is the owner, by assignment, of U.S. Patent No. 7,516,190. A true copy of the '190 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 3.

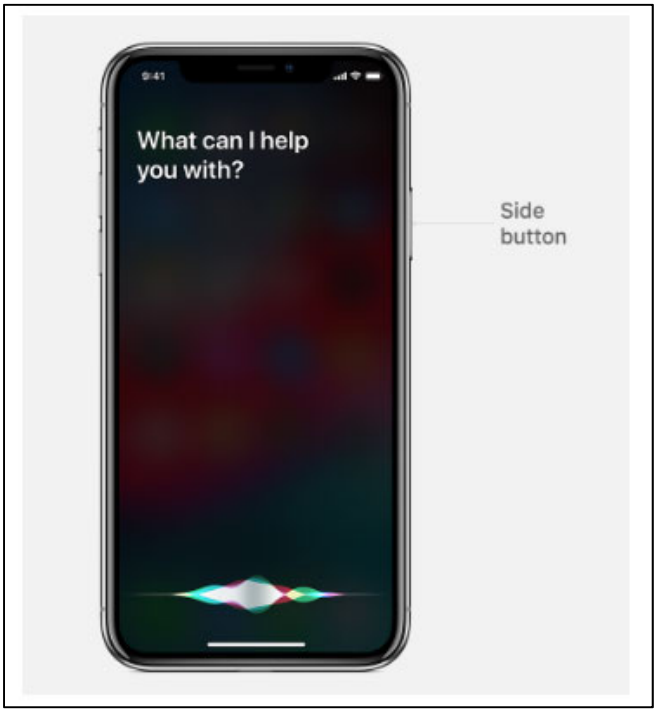
75. Defendant Apple has directly infringed, and is continuing to directly infringe, literally or under the doctrine of equivalents, at least independent claim 1 of Parus's '190 Patent by making, using, selling, and/or offering for sale its Apple devices with Siri in the United States, in violation of 35 U.S.C. § 271(a).

76. Upon filing of the complaint or shortly thereafter, Defendant Apple has knowledge of the '190 Patent.

77. Various products with Siri made or sold by Apple directly infringe at least independent claim 2 of the '705 Patent. Those Apple products include at least the Apple iPhone 6s or later models, iPad Pro 12.9 inch (3rd Generation), iPad Pro 11-inch, iPad Pro 12.9-inch (2nd Generation), iPad Pro 10.5 inch, iPad Pro 9.7 inch, iPad (6th Generation), all Apple iWatches, all HomePods, CarPlay, MacBook Pro (15 inch, 2018), MacBook Pro (13-inch, 2018, Four

Thunderbolt 3 Ports), MacBook Air (Retina, 13-inch, 2018), and iMac Pro. (Apple Accused Products). *See e.g.*, https://support.apple.com/en-us/HT209014.

78. Each of the Apple Accused Products in conjunction with Siri perform a method for allowing users to use speech commands to obtain information from a pre-defined portion of a pre-selected web site in audio format as required by claim 1 of the '190 patent. As a way of illustration, the Apple iPhone X with Siri is a voice enabled device that allows users to utter speech commands into a voice enabled device and provide users with retrieved information from pre-selected web sites in an audio form via said voice enabled device:



See e.g., https://support.apple.com/en-us/HT204389.

79. In providing the Apple Accused Products in conjunction with Siri, Apple provides a computer having a speech processor, said computer being operatively connected to the internet and to at least one phone. For example, the Apple Accused Products in conjunction with Google Assistant include at least one computer.

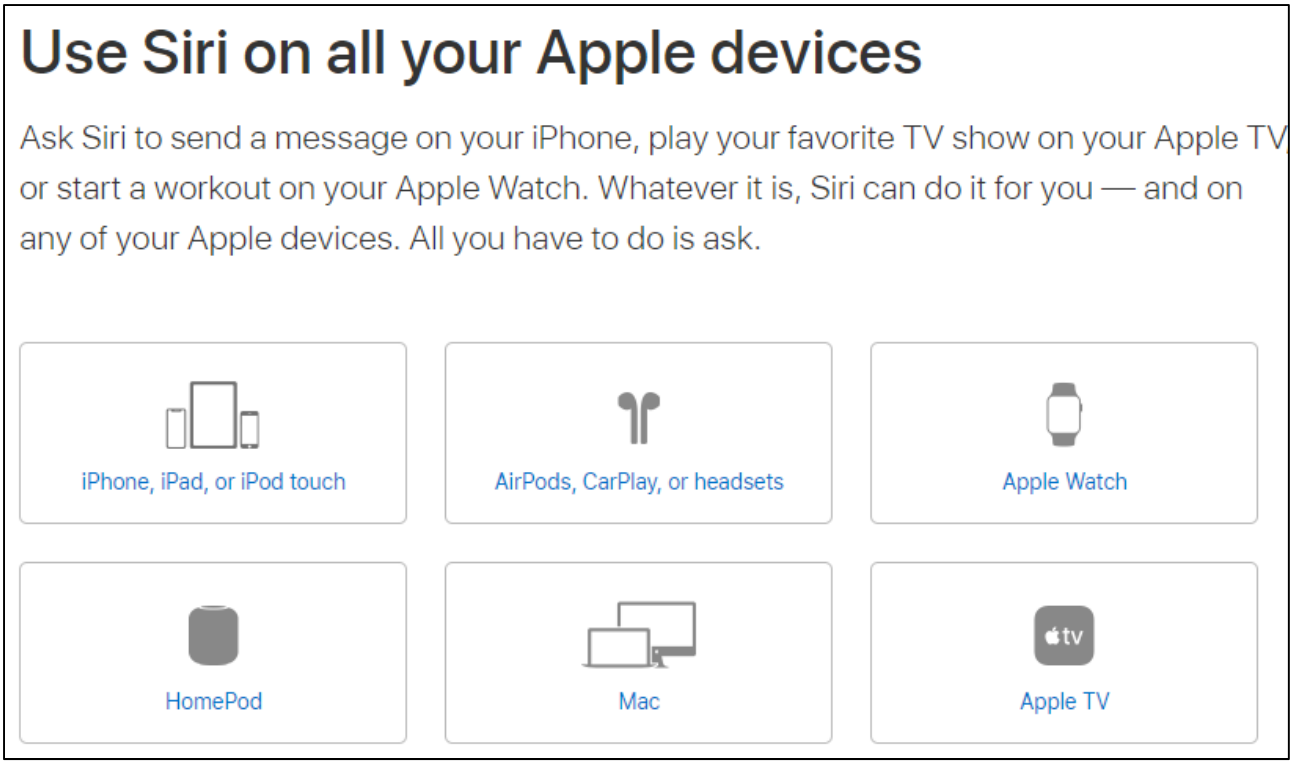
80. For example, the Apple iPhone X has a computer (i.e., microprocessor) in the A11 integrated circuit. https://support.apple.com/kb/sp770?locale=en_US. The A11 integrated circuit is operatively coupled to the internet. *Id.* As an additional example, and alternatively, the Apple iPhone X with built-in Siri has access to computing hardware that processes questions asked of Siri including 32 powerful HP servers with a total of 1024 cores and 32 terabytes of RAM a piece[4]. Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM. The company also says its text-to-speech can run on *"both general and* special purpose microprocessors, and any one or more processors of any kind of digital computer," indicating high sophistication and optimization.

> This information allegedly comes — albeit second-hand — from Apple's lead cloud architect, who says that every instance of Siri runs on 32 powerful HP servers with a total of 1024 cores and 32 terrabytes of RAM apiece. That certainly makes the new Mac Pro look long in the tooth.
>
> Specifically, each instance of Siri is made up of 4 HP c7k enclosures made up of 8 HP server blades each, with memory upgrades to 1TB of RAM.

See e.g., https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.

81. Further, the computer is operatively connected to the internet and operatively connected to the cloud:

[4] https://www.cultofmac.com/264381/hardware-siri-runs-puts-new-mac-pro-shame/.



See e.g., https://support.apple.com/en-us/HT204389.

82. Further, the Apple Accused Products include a computer have in a speech processor on the device itself and/or in the cloud.

> Siri is a personal assistant that communicates using speech synthesis. Starting in iOS 10 and continuing with new features in iOS 11, we base Siri voices on deep learning. The resulting voices are more natural, smoother, and allow Siri's personality to shine through. This article presents more details about the deep learning based technology behind Siri's voice.

See e.g., https://machinelearning.apple.com/2017/08/06/siri-voices.html.

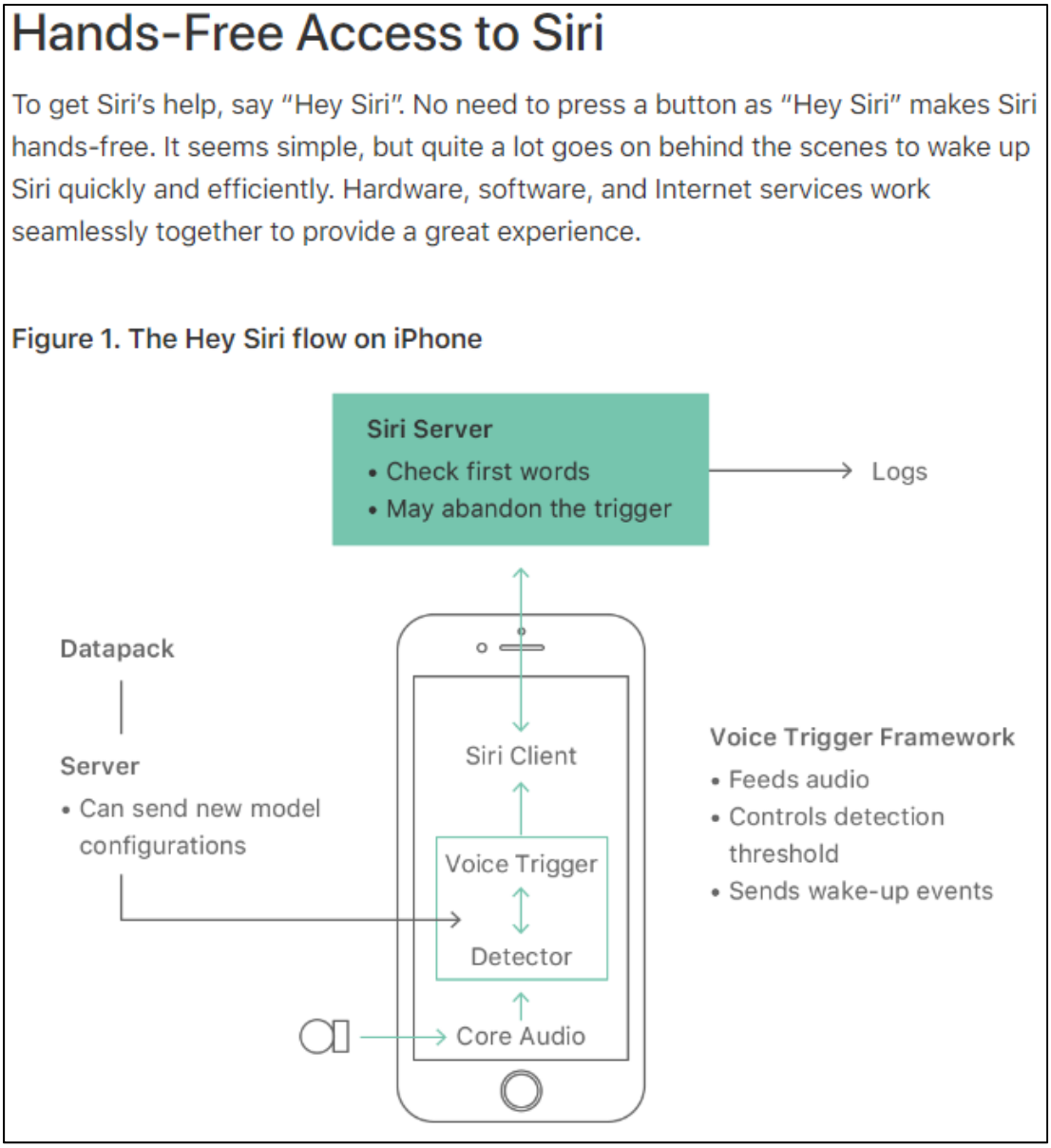
AI ON THE DEVICE

Like its rivals, Apple carries out a lot of fancy processing and machine learning tasks on data the user speaks or types. The majority of it–especially tasks that involve very personal information–happens on the device, locked away from the view of Apple or anyone else but the user.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

When you speak a request to Siri, your iPhone's software strips the request of any reference to the User ID and gives a random request ID. It's then encrypted and sent up to the cloud for more voice recognition to identify the words, and for natural language processing to understand the meaning of the words.

See e.g., https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy



See e.g., https://machinelearning.apple.com/2017/10/01/hey-siri.html.

83. Additionally, the Apple Accused Products in conjunction with Siri includes both a microphone. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/.

84. Apple provides a URL to said computer, said URL indicating a pre-selected web site from which the information is to be retrieved. For example, the Apple iPhone X includes a plurality of pre-selected web site addresses, and each pre-selected web site address identifying a

web site where the information may be retrieved. These websites have previously been crawled and pre-selected by Applebot:

> Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.
>
> > *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.

> **About Applebot**
>
> Learn about Applebot, the web crawler for Apple.
>
> Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

85. Apple uses the computer in the Apple Accused Products in conjunction with Siri to designate a pre-defined portion of the pre-selected web site which contains the information to be retrieved and to identify a named object associated with the content of the information to be retrieved:

> Apple is testing a new type of search result format for some questions. As spotted on Reddit, Spotlight Search can now answer some typed-in questions with direct answers, rather than a generic list of web results.
>
> The listing highlights the extract, which should hopefully answer the question, along with the source website and a button to report a concern if the result is inappropriate.

See, e.g., https://9to5mac.com/2019/08/27/spotlight-siri-web-answer/.

86. On information and belief, Apple uses the computer in the Apple Accused Products in conjunction with Siri to generate a regular expression based on said pre-defined portion of said pre-selected web site and said named object, said regular expression corresponding to said content of said information to be retrieved, wherein said regular expression is a text string used for describing a search pattern. For example, Siri provides a specific Web Answer that is directly pulled from the web.

> Fortunately, it looks as though a big change may be on the way to improve this situation. First noticed by users on Reddit, it appears as though questions typed into Spotlight will now deliver a specific "Web Answer" result from a wide variety of sites, along with a "Report a Concern" link.
>
> The same thing happens with Siri—ask a general question and you're now far more likely to get a direct answer pulled from the web.

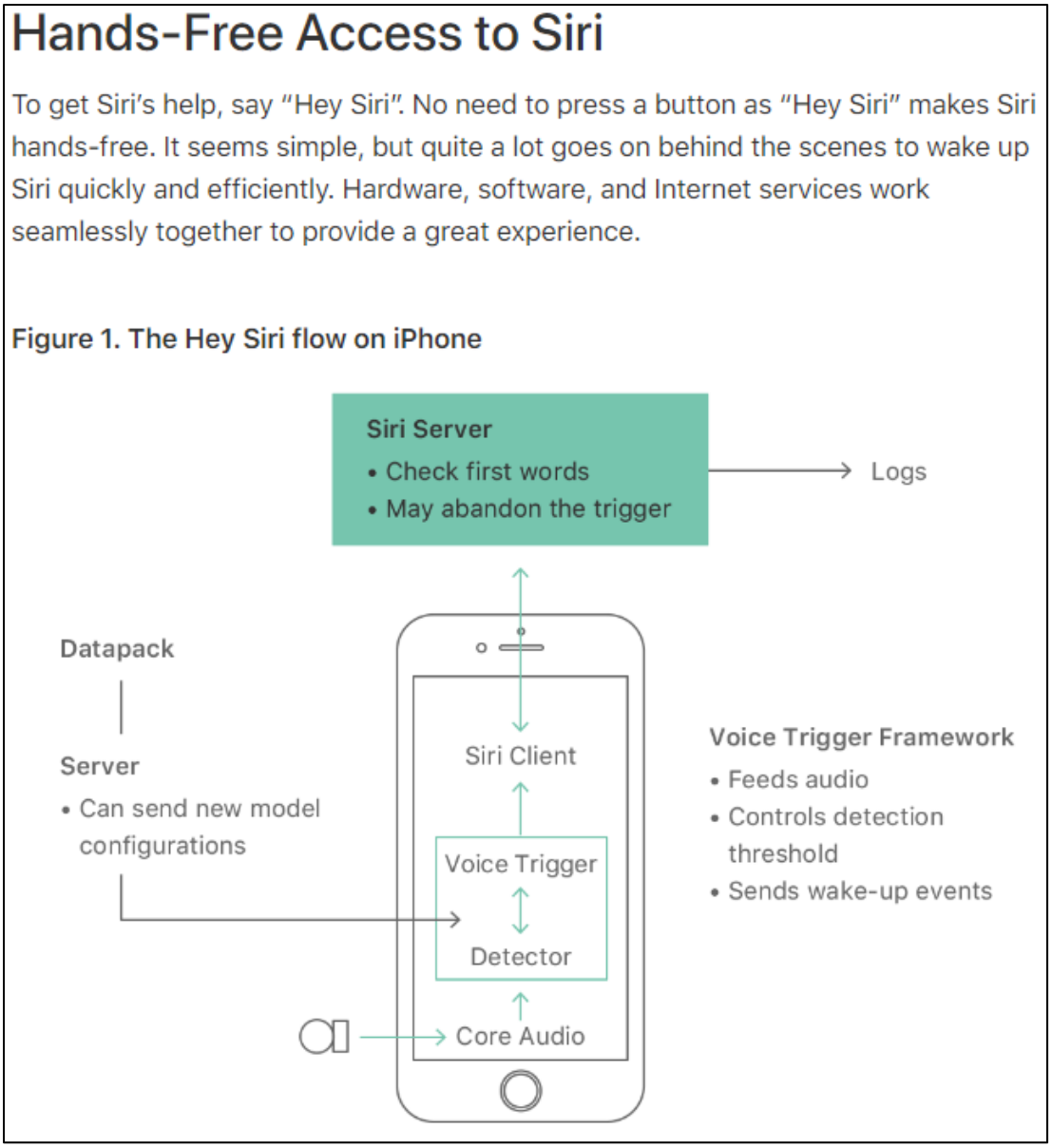
See, e.g., https://www.macworld.com/article/233195/siri-and-spotlight-now-providing-new-web-answers-to-us-users.html.

87. On information and belief, Apple provides a speech command to said speech processor, said speech command corresponding to said regular expression. For example, when a user states a speech command, Apple provides that speech command to the speaker-independent speech recognition engine, and the speech commands correspond to some regular expression, as shown in the resources cited above. *See, e.g.*, https://www.macworld.com/article/233195/siri-and-spotlight-now-providing-new-web-answers-to-us-users.html.

88. Apple uses the speech processor in the Apple Accused Products in conjunction with Siri to convert said speech command to a digital-form command.

> Siri is a personal assistant that communicates using speech synthesis. Starting in iOS 10 and continuing with new features in iOS 11, we base Siri voices on deep learning. The resulting voices are more natural, smoother, and allow Siri's personality to shine through. This article presents more details about the deep learning based technology behind Siri's voice.

See e.g., https://machinelearning.apple.com/2017/08/06/siri-voices.html.



See e.g., https://machinelearning.apple.com/2017/10/01/hey-siri.html.

89. On information and belief, Apple uses the computer in the Apple Accused Products in conjunction with Siri to receive said digital-form command from said speech processor, said computer assigning said regular expression to said digital-form command.

90. On information and belief, Apple uses the computer in the Apple Accused Products in conjunction with Siri to retrieve said regular expression corresponding to said digital-form command, as discussed above.

91. On information and belief, Apple uses the computer in the Apple Accused Products in conjunction with Siri to retrieve the information from the predefined portion of the pre-selected web site corresponding to said regular expression when the requested information is found in the pre-defined portion of the preselected website. *See, e.g.*, https://www.macworld.com/article/233195/siri-and-spotlight-now-providing-new-web-answers-to-us-users.html.

92. On information and belief, Apple uses the computer in the Apple Accused Products in conjunction with Siri to search said pre-selected web site for said named object when the requested information is not found in the pre-defined portion of the pre-selected web site.

93. Apple uses the speech processor in the Apple Accused Products in conjunction with Siri to convert said retrieved information into an audio message and Apple uses the speech processor in the Apple Accused Products in conjunction with Siri to forward said audio message to a user. *See e.g.*, https://www.macworld.co.uk/how-to/use-siri-iphone-ipad-3495151. The Apple Accused Products in conjunction with Siri includes said speech synthesis device configured to produce an audio message containing.

94. Parus has been damaged by the infringement of the '190 patent by Apple. Parus is entitled to recover from Apple the damages sustained by Parus as a result of Apple's wrongful acts.

COUNT III

APPLE'S INFRINGEMENT OF U.S. PATENT NO. 7,516,190

95. Parus restates and incorporates by reference all of the allegations made in the preceding paragraphs as though fully set forth herein.

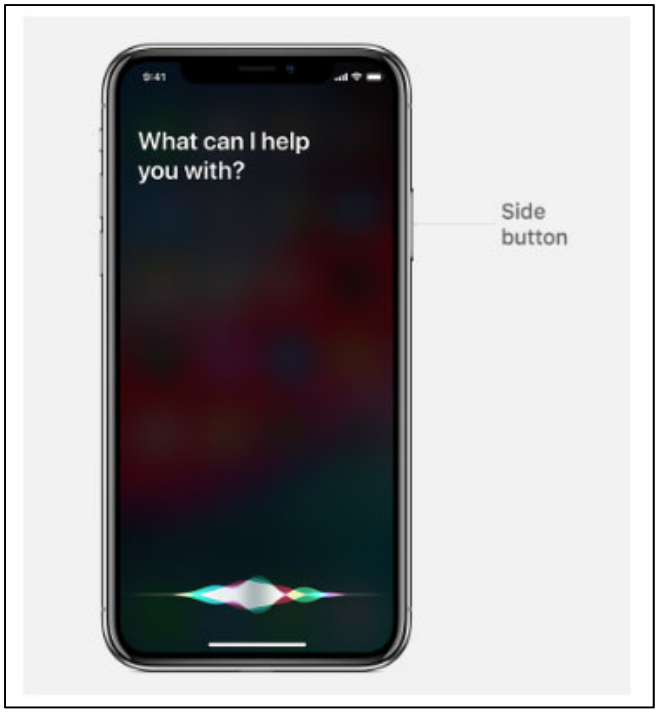
96. Parus is the owner, by assignment, of U.S. Patent No. 9,769,314. A true copy of the '314 Patent granted by the U.S. Patent & Trademark Office is attached as Exhibit 4.

97. Defendant Apple has directly infringed, and is continuing to directly infringe, literally or under the doctrine of equivalents, at least independent claim 1 of Parus's '314 Patent by making, using, selling, and/or offering for sale its Apple devices with Siri in the United States, and operating through its Google Pixel products operating the Android operating system in violation of 35 U.S.C. § 271(a).

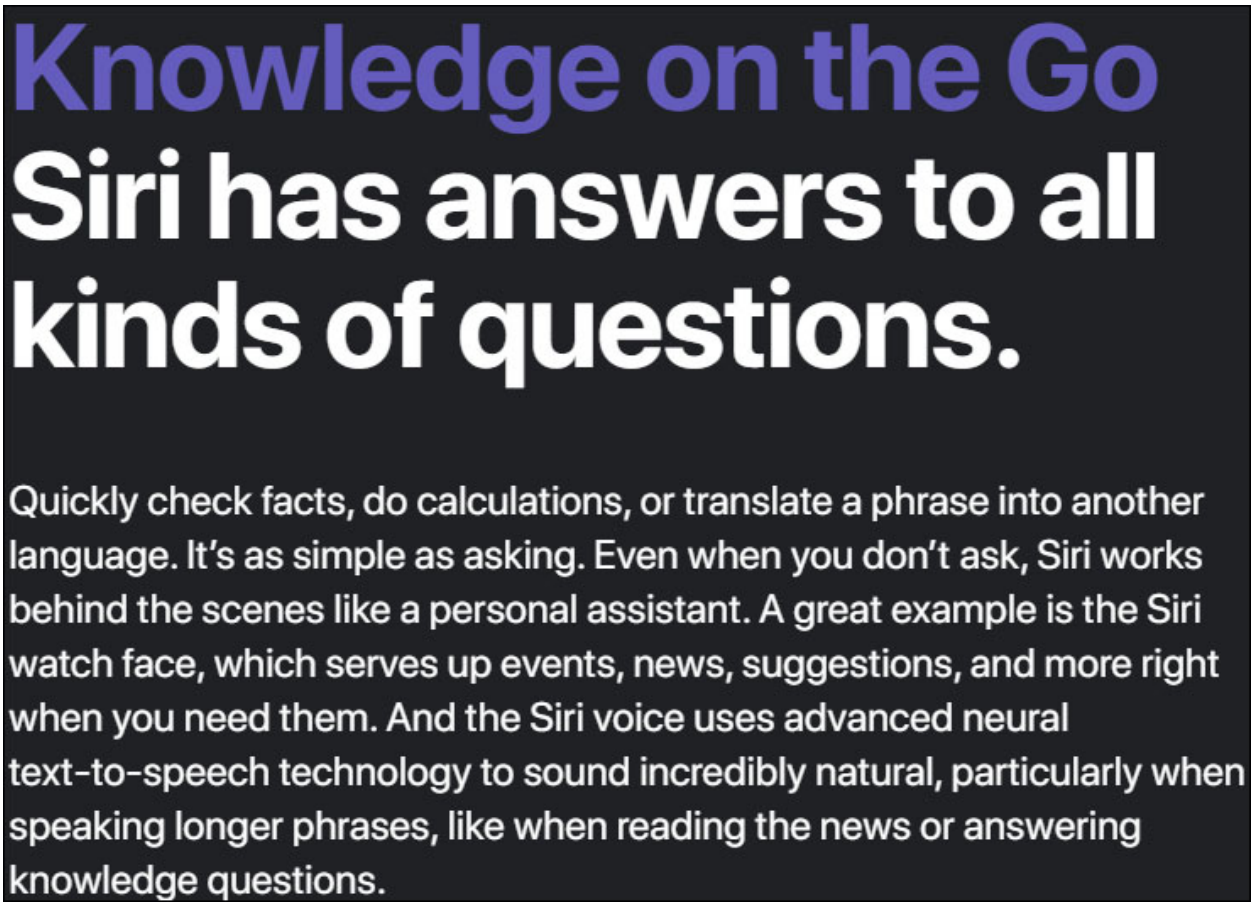
98. At least as of the filing of the original complaint, Defendant Apple has knowledge of the '314 Patent.

99. Various products with Siri made or sold by Apple directly infringe at least independent claim 1 of the '314 Patent. Those Apple products include at least the Apple iPhone X or later models, the Apple iPhone 6s or later models, iPad Pro 12.9 inch (3rd Generation), iPad Pro 11-inch, iPad Pro 12.9-inch (2nd Generation), iPad Pro 10.5 inch, iPad Pro 9.7 inch, iPad (6th Generation), all Apple iWatches, all HomePods, CarPlay, MacBook Pro (15 inch, 2018), MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports), MacBook Air (Retina, 13-inch, 2018), and iMac Pro. (Apple Accused Products). *See e.g.*, https://support.apple.com/en-us/HT209014.

100. The Apple Accused Products in conjunction with Siri perform a method for retrieving information from an information source, the information source being periodically updated with current information, over a network, by speech commands received from a particular user of a plurality of users provided by the particular user via an electronic-communication device, and wherein each of the plurality of users has a respective electronic-communication device.



See e.g., https://support.apple.com/en-us/HT204389.

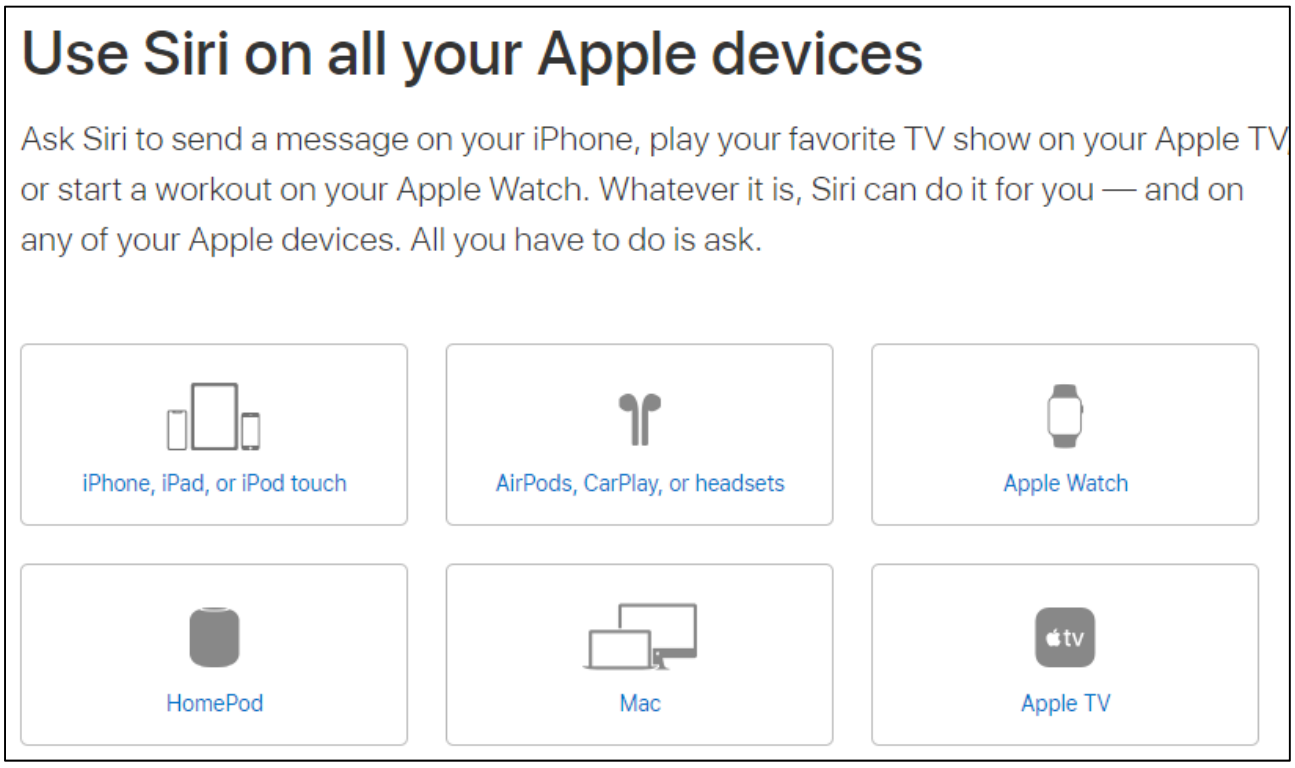


See, e.g, https://www.apple.com/siri/.

101. The Apple Accused Products receive a speech command from each of the plurality of users provided via the respective electronic-communication device, by a speech-recognition engine coupled to a media server, the media server configured to identify and access the information source via the network, the speech-recognition engine adapted to select speech-recognition grammar established to correspond to the speech commands received from the plurality of users and assigned to a desired search.

102. For example, the Apple Accused Products in conjunction with Siri are electronic-communication devices. *See, e.g.*, https://support.apple.com/en-us/HT209014.

103. Siri is built-in to Apple products including iPhones, iPads, HomePods and other devices, and include the hardware and software to process speech.



See e.g., https://support.apple.com/en-us/HT204389.

104. For example, the Apple devices in conjunction with Siri includes a microphone.

> Speak your request into the iPad or iPhone; when you've finished, the white line turns into a round microphone icon. Siri should get back to you with an answer (although sometimes it takes a few moments).

See, e.g., https://www.macworld.co.uk/how-to/use-siri-iphone-ipad-3495151/.

105. The Apple Accused Devices in conjunction with Siri are systems for retrieving information from pre-selected web sites by uttering speech commands into a voice enabled device. For example, Apple touts the Siri on its web pages. *See, e.g*, https://www.apple.com/siri/.

106. Siri retrieves information from pre-selected websites that have already been crawled by the Applebot.

> Rumors that Apple might be creating its own search engine started doing the rounds last fall, when webmasters started seeing web crawler visits from IP addresses starting with 17 – the IP address block owned entirely by Apple. Apple has now officially confirmed in a support document that it is operating its own web crawler called Applebot.
>
> > *Applebot is the web crawler for Apple, used by products including Siri and Spotlight Suggestions. It respects customary robots.txt rules and robots meta tags. It originates in the 17.0.0.0 net block.*

See e.g., https://9to5mac.com/2015/05/06/apple-search-engine-applebot/.

> # About Applebot
>
> Learn about Applebot, the web crawler for Apple.
>
> Applebot is the web crawler for Apple. Products like Siri and Spotlight Suggestions use Applebot. It respects customary robots.txt rules and robots meta tags, and it originates in the 17.0.0.0 net block.

See e.g., https://support.apple.com/en-us/HT204683.

107. The Apple Accused Products select, by the media server, at least one information-source-retrieval instruction corresponding to the speech-recognition grammar established for a particular speech command, the at least one information source-retrieval instruction stored in a database associated with the media server and adapted to retrieve information. *See, e.g.*, https://9to5mac.com/2019/08/27/spotlight-siri-web-answer/; s*ee also* https://macobserver.com/link/smarter-spotlight-search/ ("The new web answers appear to be a further expansion of this technology. If the statistics are not readily available from the knowledge database, it seems Apple is now using algorithms to crawl websites and find possible responses to questions.").

108. Apple accesses, by a web-browsing server, a portion of the information source to retrieve information of interest requested by the particular user, by using a processor of the web-browsing server, which processor (i) performs an instruction that requests information from an identified webpage, and (ii) utilizes a content extractor within the web-browsing server to separate a portion of the information from other information, the information derived from only a portion of the webpage containing information of interest to the particular user, wherein the content extractor uses a content-descriptor file containing a description of the portion of information and wherein the content-descriptor file indicates a location of the portion of the information within the information source. *See, e.g.*, https://9to5mac.com/2019/08/27/spotlight-siri-web-answer/; s*ee also* https://macobserver.com/link/smarter-spotlight-search/

109. Apple selects, by the web-browsing server, the information of interest from the information source and retrieving only the portion of the information of interest requested by the particular user according to the at least one information-source-retrieval instruction, as discussed

in the resources discussed above. *See, e.g.*, https://9to5mac.com/2019/08/27/spotlight-siri-web-answer/; s*ee also* https://macobserver.com/link/smarter-spotlight-search/

110. Apple converts the information retrieved from the information source into an audio message by a speech synthesis engine, the speech-synthesis engine coupled to the media server and transmits the audio message to the electronic communication device of the particular user requesting information of interest to the particular user. For example, the Apple iPhone X includes the speech synthesis device configured to produce an audio message containing any retrieved information from the pre-selected web sites, and the speech synthesis device further configured to transmit said audio message to said users via said voice enabled device. Because the Apple iPhone X in conjunction with Siri can handle voice commands on the device itself or in collaboration with the cloud, there is a recognition grammar corresponding to each instruction set and corresponding speech command. *See e.g.*, https://www.macworld.co.uk/how-to/iosapps/use-siri-iphone-ipad-3495151/; https://www.fastcompany.com/40443055/apple-explains-how-its-making-siri-smart-without-endangering-user-privacy.

111. Parus has been damaged by the infringement of the ’314 patent by Apple. Parus is entitled to recover from Apple the damages sustained by Parus as a result of Apple’s wrongful acts.

PRAYER FOR RELIEF

112. Parus restates and incorporates by reference all of the allegations made in the preceding

WHEREFORE, Parus request the Court grant the relief set forth below:

A. Enter judgment that Defendant has directly infringed, and continues to directly infringe, one or more claims of the '705 Patent, '190 Patent, the '402 Patent, and/or the '314 Patent;

B. Order Defendant to account for and pay damages caused to Parus by Defendant's unlawful acts of patent infringement;

C. Award Parus the interest and costs incurred in this action; and

D. Grant Parus such other and further relief, including equitable relief, as the Court deems just and proper.

DEMAND FOR JURY TRIAL

Plaintiff demands a jury trial for all issues deemed to be triable by a jury.

Dated: November 5, 2021

Respectfully submitted,

By */s/ Michael J. McNamara*
Michael J. McNamara
Massachusetts BBO No. 665885
MMcNamara@mintz.com
Michael T. Renaud
Massachusetts BBO No. 629783
MTRenaud@mintz.com
Sean M. Casey
Massachusetts BBO No. 705197
SMCasey@mintz.com
MINTZ LEVIN COHN FERRIS GLOVSKY AND POPEO PC
One Financial Center
Boston, MA 02111
Tel: 617-542-6000
Fax: 617-542-2241
www.mintz.com

Of Counsel:

T. John Ward, Jr.
Texas State Bar No. 00794818
E-mail: jw@wsfirm.com
Andrea L. Fair
Texas State Bar No. 24078488

E-mail: andrea@wsfirm.com
Claire Abernathy Henry
Texas State Bar No. 24053063
Email: claire@wsfirm.com
Charles Everingham IV
Texas State Bar No. 00787447
Email: ce@wsfirm.com
WARD, SMITH & HILL, PLLC
1507 Bill Owens Parkway
Longview, Texas 75604
(903) 757-6400 (telephone)
(903) 757-2323 (facsimile)

Attorneys for Plaintiff *Parus Holdings Inc.*

CERTIFICATE OF SERVICE

I hereby certify that all counsel of record are being served with a copy of the foregoing document via the Court's CM/ECF system on November 5, 2021.

By */s/ Michael J. McNamara*
Michael J. McNamara

EXHIBIT 1

US006721705B2

(12) **United States Patent**
Kurganov et al.

(10) **Patent No.: US 6,721,705 B2**
(45) **Date of Patent: Apr. 13, 2004**

(54) **ROBUST VOICE BROWSER SYSTEM AND VOICE ACTIVATED DEVICE CONTROLLER**

(75) Inventors: **Alexander Kurganov**, Buffalo Grove, IL (US); **Valery Zhukoff**, Deerfield, IL (US)

(73) Assignee: **Webley Systems, Inc.**, Deerfield, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 585 days.

(21) Appl. No.: **09/776,996**

(22) Filed: **Feb. 5, 2001**

(65) **Prior Publication Data**

US 2001/0047262 A1 Nov. 29, 2001

Related U.S. Application Data

(60) Provisional application No. 60/180,344, filed on Feb. 4, 2000, and provisional application No. 60/233,068, filed on Sep. 15, 2000.

(51) **Int. Cl.**[7] **G10L 21/06**

(52) **U.S. Cl.** **704/270.1**; 704/270; 707/10; 707/101; 707/102

(58) **Field of Search** 704/270, 270.1, 704/275; 707/10, 102, 101

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,596,900	A	6/1986	Jackson	179/2 A
4,776,016	A	10/1988	Hansen	381/42
4,837,798	A	6/1989	Cohen et al.	379/88
4,907,079	A	3/1990	Turner et al.	358/84
5,086,385	A	2/1992	Launey et al.	364/188
5,195,086	A	3/1993	Baumgartner et al.	370/264
5,243,645	A	9/1993	Bissell et al.	379/211
5,325,421	A	6/1994	Hou et al.	379/67
5,327,529	A	7/1994	Fults et al.	395/155
5,333,266	A	7/1994	Boaz et al.	709/206
5,475,791	A	12/1995	Schalk et al.	395/2.42
5,497,373	A	3/1996	Hulen et al.	370/79
5,499,288	A	3/1996	Hunt et al.	379/88
5,517,558	A	5/1996	Schalk	379/88
5,603,031	A	2/1997	White et al.	709/317
5,611,031	A	3/1997	Hertzfeld et al.	345/433
5,652,789	A	7/1997	Miner et al.	379/201
5,675,507	A	10/1997	Bobo, II	364/514 R
5,689,669	A	11/1997	Lynch et al.	345/355
5,719,921	A	2/1998	Vysotsky et al.	379/88
5,742,905	A	4/1998	Pepe et al.	455/461
5,752,191	A	5/1998	Fuller et al.	455/445
5,809,282	A	9/1998	Cooper et al.	395/500
5,867,494	A	2/1999	Krishnaswamy et al.	370/352
5,867,495	A	2/1999	Elliott et al.	370/352

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO	96/09710	3/1996
WO	98/23058	5/1998

OTHER PUBLICATIONS

Don Hunt and Brian Edwards "*Long–Distance Remote Control to the Rescue*" Chicago Tribune, Jun. 15, 2002, Section 4, p. 15.

Ly, "Chatter A: A Conversational Telephone Agent," *submitted to Program in Media Arts and Sciences*, Massachusetts Institute of Technology, (1993), pp. 1–130.

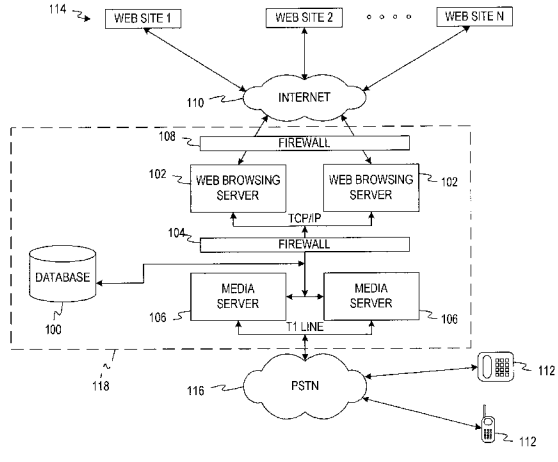
(List continued on next page.)

Primary Examiner—Susan McFadden
(74) *Attorney, Agent, or Firm*—Jenkens & Gilchrist

(57) **ABSTRACT**

The present invention relates to a system for acquiring information from sources on a network, such as the Internet. A voice browsing system maintains a database containing a list of information sources, such as web sites, connected to a network. Each of the information sources is assigned a rank number which is listed in the database along with the record for the information source. In response to a speech command received from a user, a network interface system accesses the information source with the highest rank number in order to retrieve information requested by the user.

4 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

5,873,080	A		2/1999	Coden et al. 707/3
5,881,134	A		3/1999	Foster et al. 379/88
5,884,262	A		3/1999	Wise et al. 704/270
5,890,123	A		3/1999	Brown et al. 704/275
5,915,001	A		6/1999	Uppaluru 379/88.22
5,974,413	A		10/1999	Beauregard et al. 707/6
5,999,525	A		12/1999	Krishnaswamy et al. ... 370/352
6,012,088	A		1/2000	Li et al. 709/219
6,014,437	A		1/2000	Acker et al. 379/219
6,018,710	A		1/2000	Wynblatt et al. 704/260
6,021,181	A		2/2000	Miner et al. 379/88.23
6,031,904	A		2/2000	An et al. 379/201
6,047,053	A		4/2000	Miner et al. 379/201
6,078,580	A		6/2000	Mandalia et al. 370/352
6,081,518	A		6/2000	Bowman-Amuah 370/352
6,091,808	A		7/2000	Wood et al. 379/201
6,115,742	A		9/2000	Franklin et al. 709/224
6,208,638	B1		3/2001	Rieley et al. 370/354
6,233,318	B1		5/2001	Picard et al. 379/88.17
6,252,944	B1		6/2001	Hansen, II et al. 379/67.1
6,285,745	B1		9/2001	Bartholomew et al. .. 379/88.17
6,446,076	B1	*	9/2002	Burkey et al. 707/102
6,477,420	B1		11/2002	Lim et al. 379/67.1
6,546,393	B1	*	4/2003	Khan 707/10

OTHER PUBLICATIONS

Schmandt et al., "Phone Slave: A Graphical Telecommunications Interface," *Proceedings of the SID* (1985), vol. 26/1, pp. 79–82.

Schmandt, "Phoneshell: the Telephone as Computer Terminal," *Proceedings of ACM Multimedia '93 Conference* (1993), pp. 373–382.

"Wildfire Communications Inc. (A)," *Harvard Business School* (Mar. 21, 1996), Publication No. 9–396–305, 22 pages.

Internet Web Page, "Wildfire Communications, Inc.," (Nov. 5, 1997) that includes the following addresses: http://www.wildfire.com (1 pg), http://www.wildfire.com/consumer-home.html (2 pgs), http://www.wildfire.com/106.html (2 pgs), http://www.wildfire.com/carrierhome.html (2 pgs), http://www.wildfire.com/sfandb.html (3 pgs), *http://www.wildfire.com/about.html* (1 pg), http://www.wildfire.com/abtmgmt.html (3 pgs), *http://www.wildfire.com/scoop.html* (2 pgs), http://www.wildfire.com/intel.html (1 pg), http://www.wildfire.com/msft.html (2 pgs).

* cited by examiner

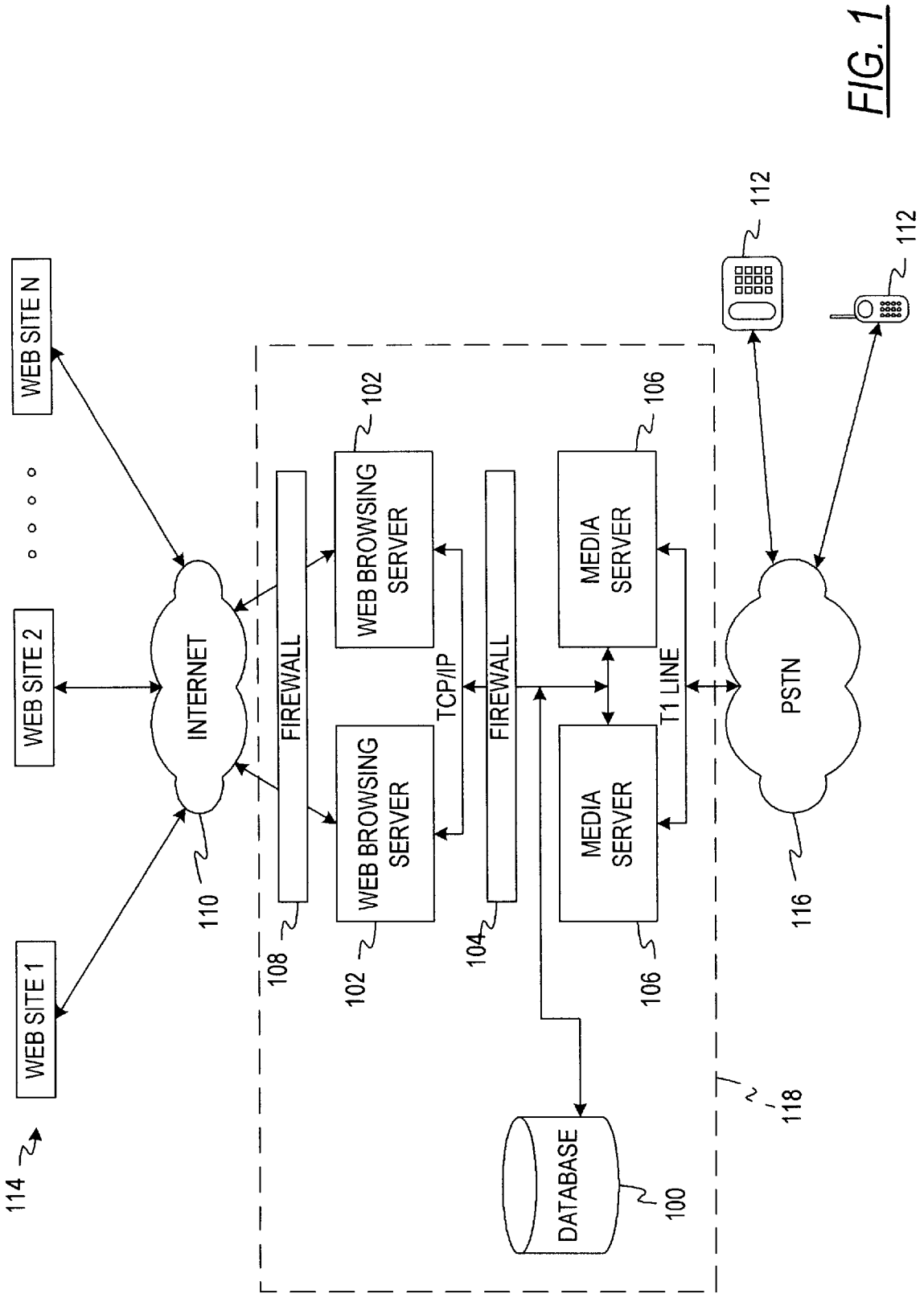

WEB SITE N
WEB SITE 2
WEB SITE 1
INTERNET
FIREWALL
WEB BROWSING SERVER
WEB BROWSING SERVER
TCP/IP
FIREWALL
MEDIA SERVER
MEDIA SERVER
T1 LINE
PSTN
DATABASE
110
108
102
102
104
106
106
112
112
116
118
100
114

FIG. 1

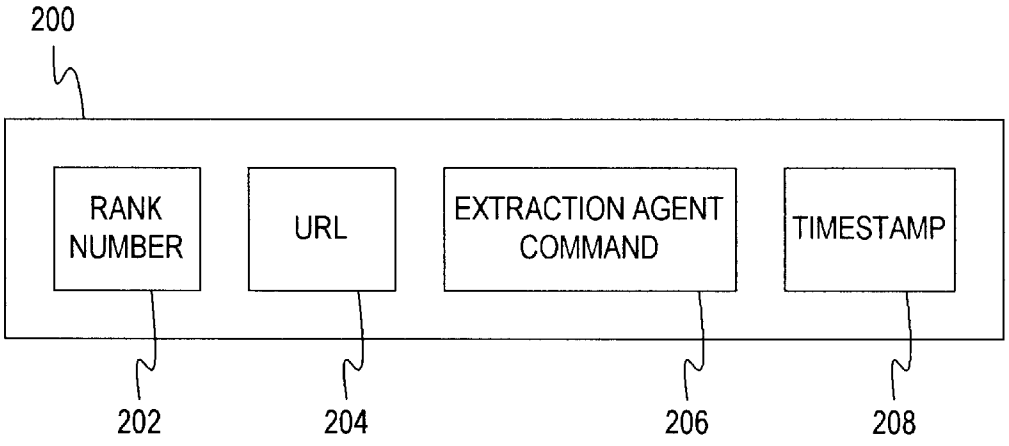

200
RANK NUMBER
URL
EXTRACTION AGENT COMMAND
TIMESTAMP
202
204
206
208

FIG. 2

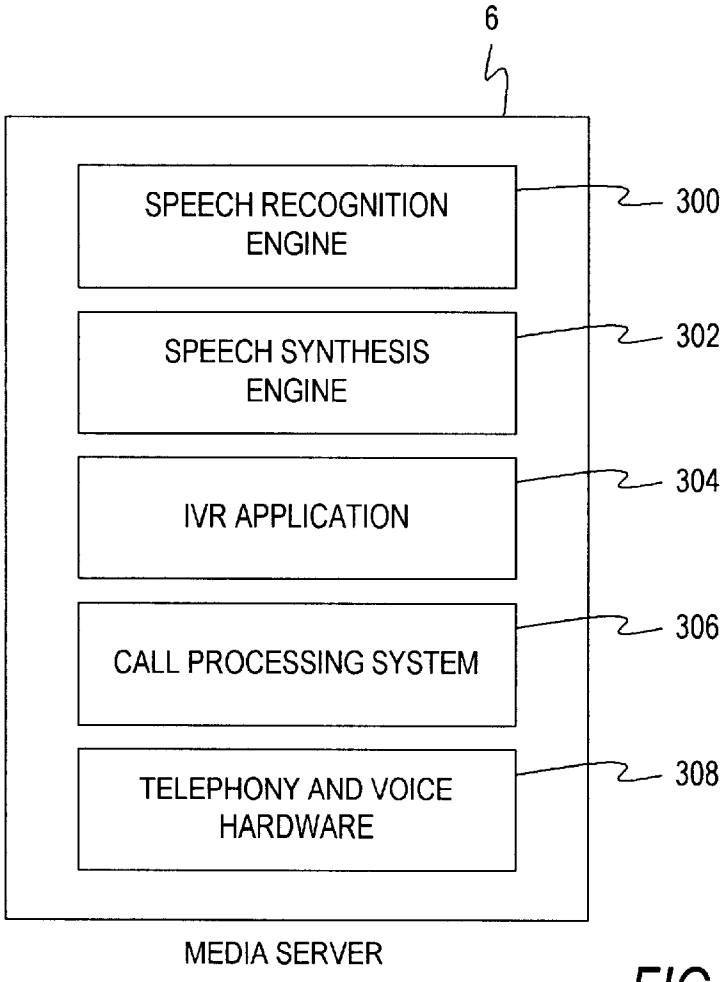

6
SPEECH RECOGNITION ENGINE
300
SPEECH SYNTHESIS ENGINE
302
IVR APPLICATION
304
CALL PROCESSING SYSTEM
306
TELEPHONY AND VOICE HARDWARE
308
MEDIA SERVER

FIG. 3

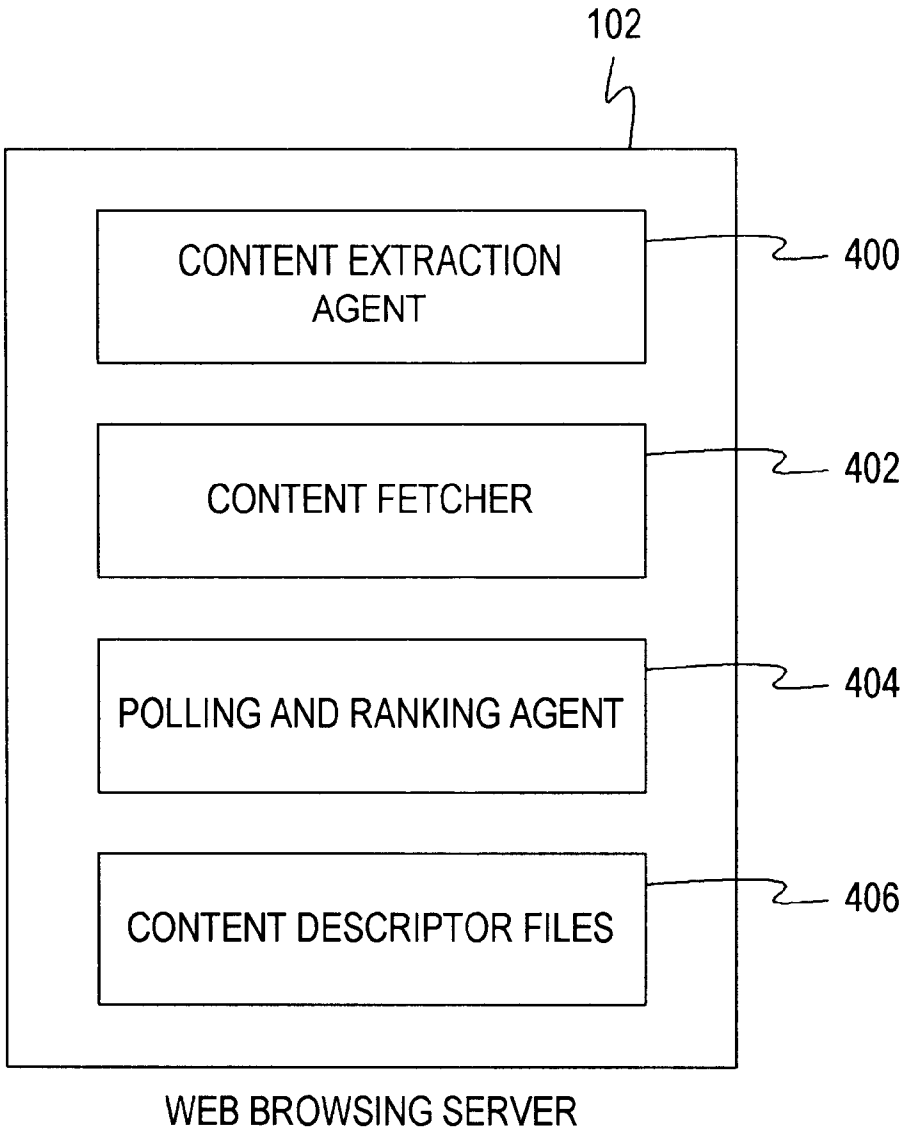

102
CONTENT EXTRACTION AGENT
400
CONTENT FETCHER
402
POLLING AND RANKING AGENT
404
CONTENT DESCRIPTOR FILES
406
WEB BROWSING SERVER

FIG. 4

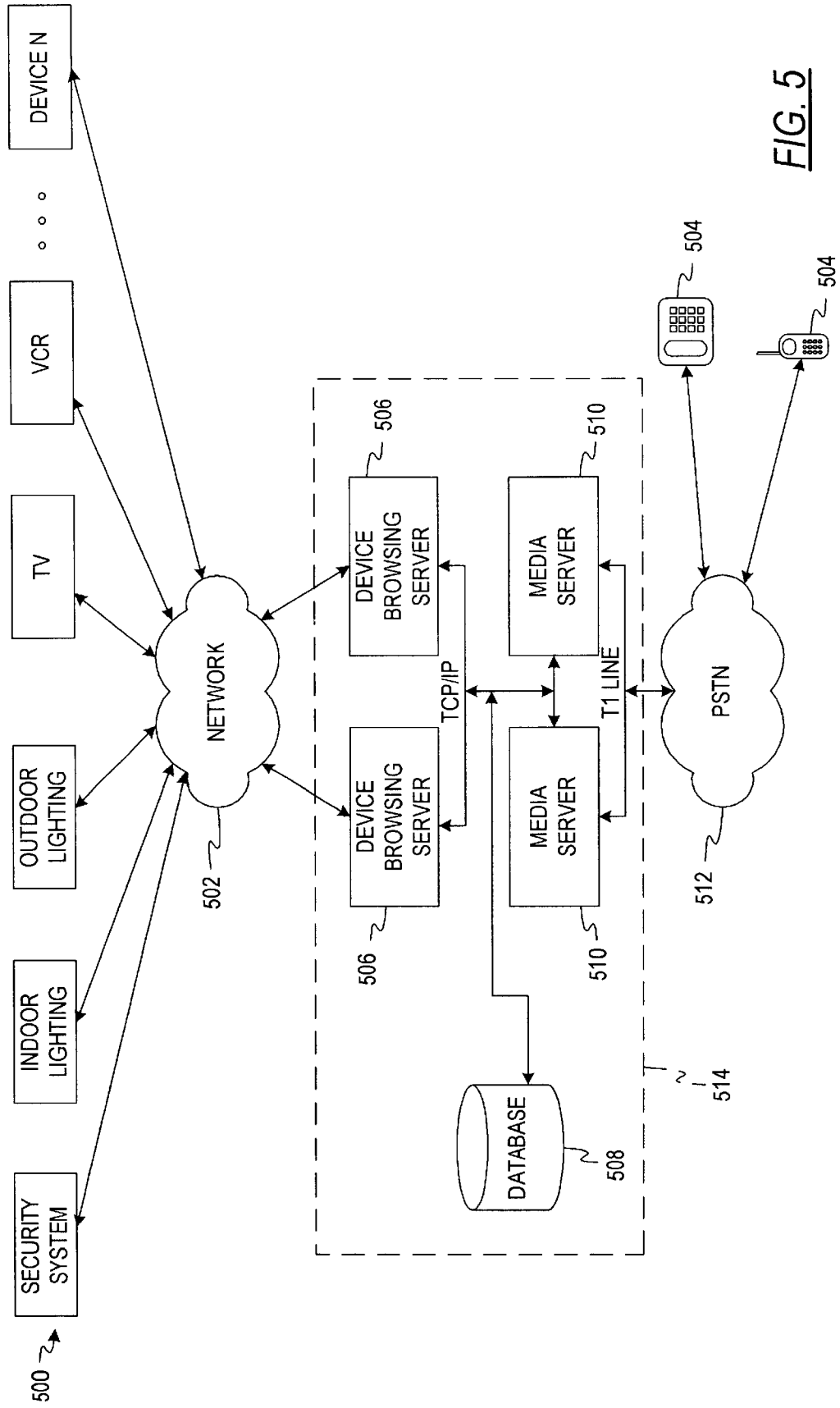

DEVICE N
VCR
TV
OUTDOOR LIGHTING
INDOOR LIGHTING
SECURITY SYSTEM
500
NETWORK
502
DEVICE BROWSING SERVER
506
DEVICE BROWSING SERVER
506
TCP/IP
MEDIA SERVER
510
MEDIA SERVER
510
T1 LINE
DATABASE
508
514
PSTN
512
504
504

FIG. 5

ROBUST VOICE BROWSER SYSTEM AND VOICE ACTIVATED DEVICE CONTROLLER

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority from U.S. Provisional Application Ser. No. 60/180,344, filed Feb. 4, 2000 entitled "Voice-Activated Information Retrieval System" and U.S. Provisional Application Ser. No. 60/233,068, filed Sep. 15, 2000 entitled "Robust Voice Browser System and Voice Activated Device Controller".

FIELD OF THE INVENTION

The present invention relates to a robust and highly reliable system that allows users to browse web sites and retrieve information by using conversational voice commands. Additionally, the present invention allows users to control and monitor other systems and devices that are connected the Internet or any other network by using voice commands.

BACKGROUND OF THE INVENTION

Currently, three options exist for a user who wishes to gather information from a web site accessible over the Internet. The first option is to use a desktop or a laptop computer connected to a telephone line via a modem or connected to a network with Internet access. The second option is to use a Personal Digital Assistant (PDA) that has the capability of connecting to the Internet either through a modem or a wireless connection. The third option is to use one of the newly designed web-phones or web-pagers that are now being offered on the market. Although each of these options provide a user with access to the Internet to browse web sites, each of them have their own drawbacks.

Desktop computers are very large and bulky and are difficult to transport. Laptop computers solve this inconvenience, but many are still quite heavy and are inconvenient to carry. Further, laptop computers cannot be carried and used everywhere a user travels. For instance, if a user wishes to obtain information from a remote location where no electricity or communication lines are installed, it would be nearly impossible to use a laptop computer Oftentimes, information is needed on an immediate basis where a computer is not accessible. Furthermore, the use of laptop or desktop computers to access the Internet requires either a direct or a dial-up connection tan an Internet Service Provider (ISP). Oftentimes, such connections are not available when a user desires to connect to the Internet to acquire information.

The second option for remotely accessing web sites is the use of PDAs These devices also have their own set of drawbacks. First, PDAs with the ability to connect to the Internet and access web sites are not readily available. As a result, these PDAs tend to be very expensive. Furthermore, users are usually required to pay a special service fee to enable the web browsing feature of the PDA. A further disadvantage of these PDAs is that web sites must be specifically designed to allow these devices to access information on the web site. Therefore, a limited number of web sites are available that are accessible by these web-enabled PDAs. Finally, it is very common today for users to carry cell phones, however, users must also carry a separate PDA if they require the ability to gather information from various web sites Users are therefore subjected to added expenses since they must pay for both cellular telephone service and also for the web-enabling service for the PDA. This results in a very expensive alternative for the consumer.

The third alternative mentioned above is the use of web-phones or web-pagers. These devices suffer many of the same drawbacks as PDAs. First, these devices are expensive to purchase. Further, the number of web sites accessible to these devices is limited since web sites must be specifically designed to allow access by these devices. Furthermore, users are often required to pay an additional fee in order to gain wireless web access Again, this service is expensive. Another drawback of these web-phones or web-pagers is that as technology develops, the methods used by the various web sites to allow access by these devices may change. These changes may require users to purchase new web-phones or web-pagers or have the current device serviced in order to upgrade the firmware or operating system stored within the device. At the least, this would be inconvenient to users and may actually be quite expensive.

Therefore, a need exists for a system that allows users to easily access and browse the Internet from any location. Such a system would only require users to have access to any type of telephone and would not require users to subscribe to multiple services.

In the rapidly changing area of Internet applications, web sites change frequently. The design of the web site may change, the information required by the web site in order to perform searches may change, and the method of reporting search results may change. Web browsing applications that submit search requests and interpret responses from these web sites based upon expected formats will produce errors and useless responses when such changes occur. Therefore, a need exists for a system that can detect modifications to web sites and adapt to such changes in order to quickly and accurately provide the information requested by a user through a voice enabled device, such as a telephone.

When users access web sites using devices such as personal computers, delays in receiving responses are tolerated and are even expected, however, such delays are not expected when a user communicates with a telephone. Users expect communications over a telephone to occur immediately with a minimal amount of delay time A user attempting to find information using a telephone expects immediate responses to his search requests. A system that introduces too much delay between the time a user makes a request and the time of response will not be tolerated by users and will lose its usefulness. Therefore, it is important that a voice browsing system that uses telephonic communications selects web sites that provide rapid responses since speed is an important factor for maintaining the system's desirability and usability. Therefore, a need exists for a system that accesses web sites based upon their speed of operation.

SUMMARY OF THE INVENTION

It is an object of an embodiment of the present invention to allow users to gather information from web sites by using voice enabled devices, such as wireline or wireless telephones.

An additional object of an embodiment of the present invention is to provide a system and method that allows the searching and retrieving of publicly available information by controlling a web browsing server using naturally spoken voice commands.

It is an object of another embodiment of the present invention to provide a robust voice browsing system that can obtain the same information from several web sites based upon a ranking order. The ranking order is automatically

adjusted if the system detects that a given web site is not functioning, is too slow, or has been modified in such a way that the requested information cannot be retrieved any longer.

A still further object of an embodiment of the present invention is to allow users to gather information from web sites from any location where a telephonic connection can be made.

Another object of an embodiment of the present invention is to allows users to browse web sites on the Internet using conversational voice commands spoken into wireless or wireline telephones or other voice enabled devices.

An additional object an embodiment of the present invention is to provide a system and method for using voice commands to control and monitor devices connected to a network.

It is an object of an embodiment of the present invention to provide a system and method which allows devices connected to a network to be controlled by conversational voice commands spoken into any voice enabled device interconnected with the same network.

The present invention relates to a system for acquiring information from sources on a network, such as the Internet. A voice browsing system maintains a database containing a list of information sources, such as web sites, connected to a network. Each of the information sources is assigned a rank number which is listed in the database along with the record for the information source. In response to a speech command received from a user, a network interface system accesses the information source with the highest rank number in order to retrieve information requested by the user.

The a preferred embodiment of the present invention allows users to access and browse web sites when they do not have access to computers with Internet access. This is accomplished by providing a voice browsing system and method that allows users to browse web sites using conversational voice commands spoken into any type of voice enabled device (i.e., any type of wireline or wireless telephone, IP phone, wireless PDA, or other wireless device). These spoken commands are then converted into data messages by a speech recognition software engine running on a user interface system. These data messages are then sent to and processed by a network interface system. This network interface system then generates the proper requests that are transmitted to the desired web site over the Internet. Responses sent from the web site are received and processed by the network interface system and then converted into an audio message via a speech synthesis engine or a pre-recorded audio concatenation application and finally transmitted to the user's voice enabled device.

A preferred embodiment of the voice browser system and method uses a web site polling and ranking methodology that allows the system to detect changes in web sites and adapt to those changes in real-time. This enables the voice browser system of a preferred embodiment to deliver highly reliable information to users over any voice enabled device. This ranking system also enables the present invention to provide rapid responses to user requests. Long delays before receiving responses to requests are not tolerated by users of voice-based systems, such as telephones When a user speaks into a telephone, an almost immediate response is expected This expectation does not exist for non-voice communications, such as email transmissions or accessing a web site using a personal computer. In such situations, a reasonable amount of transmission delay is acceptable. The ranking system of implemented by a preferred embodiment of the present invention ensures users will always receive the fastest possible response to their request.

An alternative embodiment of the present invention allows users to control and monitor the operation of a variety of household devices connected to a network using speech commands spoken into a voice enabled device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a depiction of the voice browsing system of the first embodiment of the present invention;

FIG. **2** is a block diagram of a database record used by the first preferred embodiment of the present invention;

FIG. **3** is a block diagram of a media server used by the preferred embodiment;

FIG. **4** is a block diagram of a web browsing server used by the preferred embodiment; and

FIG. **5** is a depiction of the device browsing system of the second embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A first embodiment of the present invention is a system and method for allowing users to browse information sources, such as web sites, by using naturally spoken, conversational voice commands spoken into a voice enabled device. Users are not required to learn a special language or command set in order to communicate with the voice browsing system of the present invention. Common and ordinary commands and phrases are all that is required for a user to operate the voice browsing system The voice browsing system recognizes naturally spoken voice commands and is speaker-independent, it does not have to be trained to recognize the voice patterns of each individual user. Such speech recognition systems use phonemes to recognize spoken words and not predefined voice patterns.

The first embodiment allows users to select from various categories of information and to search those categories for desired data by using conversational voice commands. The voice browsing system of the first preferred embodiment includes a user interface system referred to as a media server. The media server contains a speech recognition software engine. This speech recognition engine is used to recognize natural, conversational-voice commands spoken by the user and converts them into data messages based on the available recognition grammar. These data messages are then sent to a network interface system. In the first preferred embodiment, the network interface system is referred to as a web browsing server. The web browsing server then accesses the appropriate information source, such as a web site, to gather information requested by the user.

Responses received from the information sources are then transferred to the media server where speech synthesis engine converts the responses into audio messages that are transmitted to the user. A more detailed description of this embodiment will now be provided.

Referring to FIG. **1**, a database **100** designed by Webley Systems Incorporated is connected to one or more web browsing servers **102** as well as to one or more media servers **106**. The database may store information on magnetic media, such as a hard disk drive, or it may store information via other widely acceptable methods for storing data, such as optical disks. The database **100** contains a separate set of records for each web site accessible by the system. An example of a web site record is shown in FIG. **2**. Each web site record **200** contains the rank number of the

web site **202**, the associated Uniform Resource Locator (URL) **204**, and a command that enables the appropriate "extraction agent" **206** that is required in order to generate proper requests sent to and to format data received from the web site. The database record **200** also contains the timestamp **208** indicating the last time the web site was accessed. The extraction agent is described in more detail below. The database **100** categorizes each database record **200** according to the type of information provided by each web site. For instance, a first category of database records **200** may correspond to web sites that provide "weather" information. The database **100** may also contain a second category of records **200** for web sites that provide "stock" information. These categories may be further divided into subcategories. For instance, the "weather" category may contain subcategories depending upon type of weather information available to a user, such as "current weather" or "extended forecast". Within the "extended forecast" subcategory, a list of web site records may be stored that provide weather information for multiple days. The use of subcategories may allow the web browsing feature to provide more accurate, relevant, and up-to-date information to the user by accessing the most relevant web site. The number of records contained in each category or subcategory is not limited. In the preferred embodiment, three web site records are provided for each category.

Table 1 below depicts two database records **200** that are used with the preferred embodiment. These records also contain a field indicating the "category" of the record, which is "weather" in each of these examples.

TABLE 1

```
category: weather
URL: URL=http://cgi.cnn.com/cgi-
bin/weather/redirect?zip=_zip
rank: 1
command: web_dispatch.p1 weather_cnn
browsingServer: wportall
browsingServerBackup: wporta12
dateTime: Dec 21 2000 2:15PM
category: weather
URL: URL=http://weather.lycos.com/wcfiveday.asp?city=zip
rank: 2
command: web_dispatch.p1 weather_lycos
browsingServer: wportal1
browsingServerBackup: wportal2
dateTime: Dec 21 2000 1:45PM
```

The database also contains a listing of pre-recorded audio files used to create concatenated phrases and sentences Further, database **100** may contain customer profile information, system activity reports, and any other data or software servers necessary for the testing or administration of the voice browsing system.

The operation of the media servers **106** will now be discussed in relation to FIG. **3**. The media servers **106** function as user interface systems. In the preferred embodiment, the media servers **106** contain a speech recognition engine **300**, a speech synthesis engine **302**, an Interactive Voice Response (IVR) application **304**, a call processing system **306**, and telephony and voice hardware **308** required to communicate with the Public Switched Telephone Network (PSTN) **116**. In the preferred embodiment, each media server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The speech recognition function is performed by a speech recognition engine **300** that converts voice commands received from the user's voice enabled device **112** (i.e., any type of wireline or wireless telephone, Internet Protocol (IP) phones, or other special wireless units) into data messages. In the preferred embodiment, voice commands and audio messages are transmitted using the PSTN **116** and data is transmitted using the TCP/IP communications protocol. However, one skilled in the art would recognize that other transmission protocols may be used for either voice or data. Other possible transmission protocols would include SIP/VoIP (Session Initiation Protocol/Voice over IP), Asynchronous Transfer Mode (ATM) and Frame Relay. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www.nuance.com). The Nuance engine capacity is measured in recognition units based on CPU type as defined in the vendor specification The natural speech recognition grammars (i.e., what a user can say that will be recognized by the speech recognition engine) were developed by Webley Systems.

Table 2 below provides a partial source code listing of the recognition grammars used by the speech recognition engine of the preferred embodiment for obtaining weather information

TABLE 2

```
?WHAT_IS ?the weather ?[info information report conditions]
        ?( (?like in )
        [
UScities:n
        {<param1 $n.zip> <param2 $n.city> <param3
        $n. state>}
        ( (area code) AREA_CODE:n ) {<param1 $n>}
        ( AREA_CODE:n (area code) ) {<param1 $n>}
        ( (zip ?code) ZIP_CODE:n ) {<param1 $n>}
        ( ZIP_CODE:n (zip ?code) ) {<param1 $n>}
        ]
          )
            ) {<menu 194>}
```

The media server **106** uses recognition results generated by the speech recognition engine **300** to retrieve a web site record **200** stored in the database **100** that can provide the information requested by the user. The media server **106** processes the recognition result data identifying keywords that are used to search the web site records **200** contained in the database **100** For instance, if the user's request was "What is the weather in Chicago?", the keywords "weather" and "Chicago" would be recognized. A web site record **200** with the highest rank number from the "weather" category within the database **100** would then be selected and transmitted to the web browsing server **102** along with an identifier indicating that Chicago weather is being requested.

The media servers **106** also contain a speech synthesis engine **302** that converts the data retrieved by the web browsing servers **102** into audio messags that are transmitted to the user's voice enabled device **112**. A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhsl.com).

A further description of the web browsing server **102** will be provided in relation to FIG. **4**. The web browsing servers **102** provide access to any computer network such as the Internet **110** These servers are also capable of accessing databases stored on Local Area Networks (LANs) or Wide Area Networks (WANs). The web browsing servers receive responses from web sites and extract the data requested by the user. This task is also known as "content extraction." The web browsing servers **102** also perform the task of periodically polling or "pinging" various web sites and modifying

the ranking numbers of these web sites depending upon their response and speed. This polling feature is further discussed below. The web browsing server **102** is comprised of a content extraction agent **400**, a content fetcher **402**, a polling and ranking agent **404**, and the content descriptor files **406**. Each of these are software applications and will be discussed below.

Upon receiving a web site record **200** from the database **100** in response to a user request, the web browsing server **102** invokes the "content extraction agent" command **206** contained in the record **200**. The content extraction agent **400** allows the web browsing server **102** to properly format requests and read responses provided by the web site **114** identified in the URL field **204** of the web site record **200**. Each content extraction agent command **206** invokes the content extraction agent and identifies a content description file associated with the web page identified by the URL **204**. This content description the directs the extraction agent where to extract data from the accessed web page and how to format a response to the user utilizing that data. For example, the content description for a web page providing weather information would indicate where to insert the "city" name or ZIP code in order to retrieve Chicago weather information. Additionally, the content description file for each supported URL indicates the location on the web page where the response information is provided. The extraction agent **400** uses this information to properly extract from the web page the information requested by the user.

Table 3 below contains source code for a content extraction agent 400 used by the preferred embodiment.

TABLE 3

```
#!/usr/local/www/bin/sybper15
#$Header:
/usr/local/cvsroot/webley/agents/service/web_dispatch.p1,v
1.6
#Dispatches all web requests
#http://wcorp.itn.net/cgi/flstat?carrier=ua&flight_no=155&mo
n_abbr=jul&date=
6&stamp=OhLN~PdbuuE*itn/ord, itn/cb/sprint_hd
#http://cgi.cnnfn.com/flightview/rlm?airline=amt&number300
require "config_tmp.p1";
#check parameters
die "Usage: $0 service [params]\n" if $#ARGV < 1;
#print STDERR @ARGV;
#get parameters
my ( $service, @param ) @ARGV;
#check service
my %Services = (
                    weather_cnn =>'webget.p1 weather cnn',
                    weather_lycos =>'webget.p1'
weather_lycos',
                    weather_weather => 'webqet.p1
weather_weather',
                    weather_snap => 'webget.p1
weather_snap'
                    weather infospace => 'webget.p1
weather_infospace',
                    stockQuote_yahoo => 'webget.p1 stock',
                    fliqhtStatus_itn => 'webget.p1'
flight_delay',
                    yellowPages_yahoo => 'yp_data.p1',
                    yellowPages yahoo => 'yp data.p1',
                    newsHeaders_newsreal => 'news.p1',
                    newsArticle_newsreal => 'news.p1',
# test param
my $date = 'date';
chop ( $date );
my ( $short_date ) = $date =~ /\s+(\w{3}\s+\d{1,2})\s+/;
my %Test = {
                    weather_cnn => '60053',
                    weather_lycos => '60053',
                    weather_weather => '60053',
                    weather_snap > '60053',
                    weather_infospace => '60053',
                    stockQuote_yahoo => 'msft',
                    flightStatus_itn => 'ua 155 '.
$short_date,
                    yellowPages_yahoo => 'tires 60015',
                    newsHeaders_newsreal => '1',
                    newsArticle_newsreal > '1 1',
                    );
die "$date: $0: error: no such service: $service (check this
script) \n"
unless $Services{$service };
#prepare absolute path to run other scripts
my ( $path, $script ) = $0 =~ m|^(.*/) ([^/]*) |;
#store the service to compare against datatable
my $service_stored = $service;
#run service
while( !( $response '$path$Services{$service }@param') )
{
# response failed
# check with test parameters
$response = '$path$Services{$service }$Test{$service
}';
# print "test: $path$Services{$service }$Test{$service
}';
    if ( $response ) }
      $service =&switch_service( $service );
#     print "Wrong parameter values were supplied:
$service -
@param\n";
# die "$date: $0: error: wrong parameters: $service
-
@param\n";
    }
    else {
    # change priority and notify
    $service = &increase_attempt( $service );
    }
}
# output the response
print $response;
sub increase_attempt {
    my ( $service ) =@_;
    my ( $service_name ) = split( /_/, $service );
    print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
    # update priority
    &db_query( "update mcServiceRoute "
                    ."set priority = ( select max( priority )
from
mcServiceRoute "
                    ."where service = '$service_name') + 1,"
                    ."date = getdate( ),"
                    ."attempt = attempt + 1"
                    ."where route = '$script $service'" );
# print "---$route===\n";
  # find new route
   my $route = @{&db_query( "select route from
mcServiceRoute "
                    ."where service =
'$service name'"
                    ."and attempt < 5"
                    ."order by
priority ")
                    }-> [ 0 ] {route };
    &db_query( "update mcServiceRoute "
                    ."set attempt = 0"
                    ."where route = '$script $service'" )
                    if ( $route eq "$script $service"
                    or $route eq "$script $service_stored" ) ;
    ( $service_name, $service ) =split( /\s+/, $route );
    die "$date: $0: error: no route for the service:
$service (add
more) \n"
      unless $service;
    return $service;
}
```

TABLE 3-continued

```
sub switch service {
    my ( $service ) = @_;
    my ( $service_name ) = split( /_/, $service );
    print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
    # update priority
    &db_query( "update mcServiceRoute "
                    ."set priority = ( select max( priority )
from
mcServiceRoute "
                    ."where service = '$service_name') + 1,"
                    ."date = getdate( )"
                    ."where route = '$script $service'" );
# print "---$route===\n";
    # find new route
    my $route = @{&db_query( "select route from
mcServiceRoute "
                    ."where service =
'$service name'"
                    ."and attempt < 5"
                    ."order by
priority ")
                    }-> [ 0 ] {route };
    die "$date: $0: error: there is the only service:
$route (add
more) \n"
      if ( $route eq "$script $service"
        or $route eq "$script $service_stored" );
    $service_name, $service ) = split( /\s+/, $route );
    die "$date: $0: error: no route for the service:
$service (add
more)\n"
      unless $service;
    return $service;
}
```

Table 4 below contains source code of the content fetcher **402** used with the content extraction agent **400** to retrieve information from a web site.

TABLE 4

```
#!/usr/local/www/bin/sybper15
#-T
# -w
#$Header:
/usr/local/cvsroot/webley/agents/service/webget.p1,v 1.4
# Agent to get info from the web.
# Parameters: service_name [service_parameters], i.e. stock
msft or weather
60645
# Configuration stored in files service_name.ini
# if this file is absent the configuration is received from
mcServices table
# This script provides autoupdate to datatable if the .ini
file is newer.
$debug = 1;
use URI::URL;
use LWP::UserAgent;
use HTTP::Request::Common;
use Vail::VarList;
use Sybase::CTlib;
use HTTP::Cookies;
#print "Sybase::CTlib $DB_USR, $DB_PWD, $DB_SRV;";
open( STDERR, ">>$0.log" ) if $debug;
# open( STDERR, ">&STDOUT" );
$log='date';
# $response './url.p1
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip= 60605"';
#$response = 'pwd';
#print STDERR "pwd=$response\n";
#$response '1s';
#print STDERR "1s=$response\n";
chop( $log );
$log .= "pwd=" .'pwd';
```

TABLE 4-continued

```
chop( $log );
#$debug2 = 1;
my $service = shift;
$log .= "$service:". join( ':', @ARGV ) . "\n";
print STDERR $iog if $debug;
#$response = './ur1.p1
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"';
my @ini = &read_ini( $service );
chop( @ini );
my $section = " ";
do {$section &process_section( $section ) }while $section;
#$response = './ur1.p1
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"';
exit;
##############################
sub read_ini {
    my ( $service ) = @_;
    my @ini = ( );
    # first, try to read file
    $0 = ~ m|^(.*/) [^/]*|;
    $service = $1 . $service;
    if ( open( INT, "$service.ini" ) ) {
      @ini = (<INI>);
      return @ini unless ( $DB_SRV );
      # update datatable
      my $file_time time - int( ( -M "$service.ini" )
* 24 *
3600 );
#   print "time $file_time\n";
    my $dbh = new Sybase::CTlib $DB_USR, $DB_PWD,
$DB_SRV;
    unless ( $dbh ) {
    print STDERR "webget.p1: Cannot connect to
dataserver $DB_SRV:$DB_USR:$DB_PWD\n";
    return @ini;
    }
    my @row_refs = $dbh->ct_sql( "select lastUpdate
from
mcServices where service '$service'", undef, 1 );
    if ( $dbh->{RC }== CS_FAIL ) {
    print STDERR "webget.p1: DB select from
mcServices
failed\n";
    return @ini;
    }
    unless ( defined @row_refs {
    # have to insert
    my ( @ini_escaped ) = map {
      ( my$x = $_) =~ s/\'/\'\'/g;
      $x;
    }@ini;
    $dbh->ct_sq1( "insert mcServices values(
'$service',
'@ini_escaped', $file time )");
    if ( $dbh->{RC }== CS FAIL ) {
    print STDERR "webget.p1: DB insert to
mcServices failed\n";
    }
    return @ini;
    }
#   print "time $file_time:".$row_refs[ 0 ]-
>{'lastUpdate'
}."\n";
    if ( $file_time > $row_refs[ 0 ]->{'lastUpdate' })
{
    # have to update
    my ( @ini_escaped ) = map {
      ( my $x = $_) =~ s/\'/\'\'/g;
      $x;
      }@ini;
      $dbh->ct_sq1( "update mcServices set config =
      '@ini_escaped', lastUpdate = $file_time where service =
      '$servlce'" );
```

TABLE 4-continued

```
    if ( $dbh->55 RC }== CS FAIL ) {
      print STDERR "webget.p1: DB update to
mcServices failed\n";
        }
      }
      return @ini;
    }
    else {
    print STDERR "$0: WARNING: $service.ini n/a in ".
'pwd'
      ."Try to read DB\n";
    }
    # then try to read datatable
    die "webget.p1: Unable to find service $service\n"
unless ( $DB_SRV
);
    my $dbh new Sybase::CTlib $DB_USR, $DB_PWDI $DB_SRV;
    die "webget.p1: Cannot connect to dataserver
$DB_SRV:$DB_USR:$DB_PWD\n" unless ( $dbh );
    my @row_refs $dbh->ct_sq1( "select config from
mcServices where
service = '$service'", undef, 1 );
    die "webget.p1: DB select from mcServices failed\n" if
$dbh->{RC }
== CS_FAIL;
    die "webget.p1: Unable to find service $service\n"
unless ( defined
@row_refs );
    $row_refs[ 0 ]->{'config'}=~ [] s/\n /\n\r/g;
    @ini = split( /\r/, $row_refs[ 0 ]->{'config' });
    return @ini;
}
##############################
sub process_section {
    my ( $prev_section ) = @_;
    my ( $section, $output, $content );
    my %Param;
    my %Content;
# print"################################\n";
foreach ( @ini ) {
#   print;
#   chop;
    s/\s+$//;
    s/^\s+//;
    # get section name
    if ( /^\[(.*)\]/) {
#     print "$_: $section:$prev_section\n";
    last if $section;
  #next if $1 eq "print";
# next if $prev_section ne " " and $prev_section
ne $1;
    if ( $prev_section eq $1 ) {
    $prev_section = " ";
    next;
      }
      $section = $1;
    }
    # get parameters
    push( @{$Param{$1 }}, $2 ) if $section and
/([^=] +) = (.*)/;
    }
# print"++++++++++++++++++++++++++++++++++\n";
  return 0 unless $section;
# print "section $section\n";
  #substitute parameters with values
  map {$Param{URL }->[ 0 ] =~ s/$Param{Input }->[ S_
]/$ARGV[ $_
]/g
    }0 . . . $#{$Param{Input }};
    #get page content
    ( $Content{'TIME' }, $content ) = &get_url_content(
${$Param{URL
}}[ 0 ] );
    # filter it
    map {
      if (/\"([^\"]+)\"([^\"]*)\"/ or
/\/([^\/]+)\/([^\/]*)\//)
        my $out = $2; $content s/$1/$out/g;
    }
```

TABLE 4-continued

```
}@{$Param{"Pre-filter" }};
#print STDERR $content;
    #do main regular expression
    unless( @values = $content =~
/${$Param{Regular_expression }}[ 0
}/ ) {
    &die_hard( ${$Param{Regular_expression }}[ 0 ],
$content
);
      return $section;
    }
    %Content = map {( $Param{Output }->[ $_ ], $values[ $_
])
    }0 . . . $#{$Param{Output }};
    #filter it
    map {
      if ( /([^\"]+)\"([^\"]+)\"([^\"]*)\"/
      or /([^\/]+)\/([^\/]+)\/([^\/]*)\//) {
      my $out = $3;
      $Content{$1 }=~ s/$2/$out/g;
    }
    }@{$Param{"Post-filter" }};
    # calculate it
    map {
      if ( /([^=]+)=(.*)/ ) {
      my $eval = $2;
      map {$eval =~ s/$_/$Content{$_}/g
      }keys %Content;
      $Content{$1 }= eval( $eval );
    }
    }@{$Param{Calculate }};
    # read section [print]
    foreach $1 ( 0 . . . $#ini ) {
      next unless $ini[$1 ] =~ /^\[print\]/;
      foreach($i + 1 . . . $#ini) {
        last if $ini[$_] =~ /^\[.+\]/;
      $output .= $ini[$_] . "\n";
      }
        last;
    }
      # prepare output
      map {$output =~ s/$_/$Content{$_}/g
      }keys %Content;
      print $output;
    return 0;
}
##############################
sub get_url_content {
    my ( $url ) = @_;
    print STDERR $url if $debug;
# $response = './ur1.p1 '$ur1' ';
    $response = './ur1.p1 'url' ';
    return( $time - time, $response );
    my $ua = LWP::UserAgent->new;
    $ua->agent( 'Mozilla/4.0 [en] (X11; I; FreeBSD 2.2.8-
STABLE i386)'
);
# $ua->proxy( ['http', 'https'],
'http://proxy.webley:3128/');
# $ua->no proxy('webley', 'vail');
    my $cookie = HTTP::Cookies->new;
    $ua->cookie_jar( $cookie );
    $ur1 = ur1 $ur1;
    print "$ur1\n" if $debug2;
    my $time = time;
    my $res = $ua->request ( GET $ur1 );
    print "Response: " . ( time - $time ) . "sec\n" if
$debug2;
    return( $time - time, $res->content );
}
##############################
sub die_hard {
    my( $re, $content ) = @_;
    my ( $re_end, $pattern );
    while( $content !~ /$re/ ) {
    if ( $re =~ s/(\([^\(\)]+\)[^\(\)]*$)// ) {
      $re_end = $1 . $re end;
      }
```

TABLE 4-continued

```
        else {
            $re_end = $re;
            last;
        }
    }
    $content =~ /$re/;
    print STDERR "The regular expression did not match:\n
$re\n
Possible misuse:
$re_end:\n
Matched:
$&\n
Mismatched:
$'\n
" if $debug;
    if ( $debug ) {
        print STDERR "Content:\n $content\n"unless
$, ;
    }
}
##############################
```

Table 5 below contains the content descriptor file source code for obtaining weather information from the web site www.cnn.com that is used by the extraction agent **400** of the preferred embodiment.

TABLE 5

```
[cnn]
Input= _zip
URL= http://cgi.cnn.com/cgi-bin/weather/redirect?zip= _zip
Pre-filter="\n" "
Pre-filter="<[^<>]+>""
Pre-filter=/\s+/ /
Pre-filter="[\(\)\[]"!"
Output=_location
Output=first_day_name
Output=first_day_weather
Output=first_day_high_F
Output=first_day_high_C
Output=first_day_low_F
Output=first_day_low_C
Output=second_day_name
Output=second_day_weather
Output=second_day_high_F
Output=second_day_high_C
Output=second_day_low_F
Output=second_day_low_C
Output=third_day_name
Output=third_day_weather
Output=third_day_high_F
Output=third_day_high_C
Output=third_day_low_F
Output=third_day_low_C
Output=fourth_day_name
Output=fourth_day_weather
Output=fourth_day_high_F
Output=fourth_day_high_C
Output=fourth_day_low_F
Output=fourth_day_low_C
Output=undef
Output=_current_time
Output=_current_month
Output=_current_day
Output=_current_weather
Output=_current_temperature_F
Output=_current_temperature_C
Output=_humidity
Output=_wind
Output=_pressure
Output=_sunrise
Output=_sunset
Regular_expression=Author   (.+) Four Day Forecast
(\S+) (\S+) HIGH
(\S+) F (\S+) C LOW (\S+) F (\S+) C (\S+) (\S+) HIGH (\S+) F
```

TABLE 5-continued

```
(\S+) C LOW
(\S+) F (\S+) C (\S+) (\S+) HIGH (\S+) F (\S+) C LOW (\S+) F
(\S+) C (\S+)
(\S+) HIGH (\S+) F (\S+) C LOW (\S+) F (\S+) C (.+) Current
Conditions ( .+)
!local!, (\S+) (\S+) (.+) Temp: (\S+) F, (\S+) C Rel.
Humidity: (\S+) Wind:
(.+) Pressure: (.+) Sunrise: (.+) Sunset: (.+) Related Links
Post-filter=_current_weather"p/"partly"
Post-filter=_current weather"l/"little"
Post-filter=_current_weather"m/"mostly"
Post-filter=_current_weather"t-/"thunder"
Post-filter=_wind"N"North"
Post-filter=_wind"E"East"
Post-filter=_wind"S"South"
Post-filter=_wind"W"West"
Post-filter=_wind/mph/miles per hour/
Post-filter=_wind/kph!/kilometers per hour/
Post-filter=_wind"\s+!", "
[print]
Current weather in _location is current_weather.
Temperature is current_temperature_F Fahrenheit,
_current_temperature_C
Celsium.
Humidity is _humidity.
Wind from the _wind.
```

Table 6 below contains the content descriptor file source code for obtaining weather information from the web site www.lycos com that is used by the extraction agent **400** of the preferred embodiment.

TABLE 6

```
[lycos]
Input= zip
Input= _city
URL= http://weather.lycos com/wcfiveday.asp?city=zip
Pre-filter="\n"
Pre-filter="</TD>"td"
Pre-filter="<!.*?->""
Pre-filter="<br>"_br_"
Pre-filter=/alt= "/>alt=/
Pre-filter="<[^<>]+>""
Pre-filter=" " "
Pre-filter=/\s+/ /
Output= _location
Output=_current_weather
Output=_current_temperature_F
Output=_humidity
Output=_winddir
Output=_windspeed
Output=_windmeasure
Output=_pressure
Output=first_day_name
Output=second_day_name
Output=third_day_name
Output=fourth_day_name
Output=fifth_day_name
Output=first_day_weather
Output=second_day_weather
Output=third_day_weather
Output=fourth_day_weather
Output=fifth_day_weather
Output=first_day_high_F
Output=first_day_low_F
Output=second_day_high_F
Output=second_day_low_F
Output=third_day_high_F
Output=third_day_low_F
Output=fourth_day_high_F
Output=fourth_day_low_F
Output=fifth_day_high_F
Output=fifth_day_low_F
Output=_windkmh
```

TABLE 6-continued

```
Regular_expression= Guide My Lycos (.+) Click image to
enlarge
a1t= ([^"]+)"(?:.+) Temp: (\d+) (?:.+)F __br__ Humidity:
(\S+) (?: .+) Wind: (.+?)
__br__
Output=__current__temperature__C
Post-filter=__location"__br__""
Post-filter=__current__weather"p/"partly"
Post-filter=__current__weather"m/"mostly"
Post-filter=__current__weather"t-/"thunder"
Post-filter=__winddir"@" at"
Post-filter=__winddir/mph/miles per hour/
Post-filter=__wind/kph!/kilometers per hour/
Calculate=__current__temperature__C=int ( (__current__temperature__F
-32)*5/9)
Calculate=__windkmh= int (__windspeed*1.6)
[print]
The current weather in __location is __current__weather.
The current temperature is __current__temperature__F Farenheit
__current__temperature__C Celcius.
Humidity is __humidity.
Winds __winddir.
```

Once the web browsing server **102** accesses the web site specified in the URL **204** and retrieves the requested information, the information is forwarded to the media server **106**. The media server uses the speech synthesis engine **302** to create an audio message that is then transmitted to the user's voice enabled device **112** In the preferred embodiment, each web browsing server **102** is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

Referring to FIG. **1**, the operation of the robust voice browser system will be described. A user establishes a connection between his voice enabled device **112** and a media server **106**. This may be done using the Public Switched Telephone Network (PSTN) **116** by calling a telephone number associated with the voice browsing system **118**. Once the connection is established, the media server **106** initiates an interactive voice response (IVR) application **304**. The IVR application plays audio messages to the user presenting a list of options, such as, "stock quotes", "flight status", "yellow pages", "weather", and "news". These options are based upon the available web site categories and may be modified as desired. The user selects the desired option by speaking the name of the option into the voice enabled device **112**.

As an example, if a user wishes to obtain restaurant information, he may speak into his telephone the phrase "yellow pages". The IVR application would then ask the user what he would like to find and the user may respond by stating "restaurants". The user may then be provided with further options related to searching for the desired restaurant. For instance, the user may be provided with the following restaurant options, "Mexican Restaurants", "Italian Restaurants", or "American Restaurants". The user then speaks into the telephone **112** the restaurant type of interest. The IVR application running on the media server **106** may also request additional information limiting the geographic scope of the restaurants to be reported to the user. For instance, the IVR application may ask the user to identify the zip code of the area where the restaurant should be located. The media server **106** uses the speech recognition engine **300** to interpret the speech commands received from the user. Based upon these commands, the media server **106**retrieves the appropriate web site record **200** from the database **100** This record and any additional data, which may include other necessary parameters needed to perform the user's request, are transmitted to a web browsing server **102**. A firewall **104** may be provided that separates the web browsing server **102** from the database **100** and media server **106**. The firewall provides protection to the media server and database by preventing unauthorized access in the event the firewall for web browsing server **108** fails or is compromised. Any type of firewall protection technique commonly known to one skilled in the art could be used, including packet filter, proxy server, application gateway, or circuit-level gateway techniques.

The web browsing server **102** then uses the web site record and any additional data and executes the extraction agent **400** and relevant content descriptor file **406** to retrieve the requested information.

The information received from the responding web site **114** is then processed by the web browsing server **102** according to the content descriptor file **406** retrieval by the extraction agent. This processed response is then transmitted to the media server **106** for conversion into audio messages using either the speech synthesis software **302** or selecting among a database of prerecorded voice responses contained within the database **100**.

As mentioned above, each web site record contains a rank number **202** as shown in FIG. **2**. For each category searchable by a user, the database **100** may list several web sites, each with a different rank number **202**. As an example, three different web sites may be listed as searchable under the category of "restaurants". Each of those web sites will be assigned a rank number such as 1, 2, or 3. The site with the highest rank (i.e., rank=1) will be the first web site accessed by a web browsing server **102**. If the information requested by the user cannot be found at this first web site, then the web browsing server **102** will search the second ranked web site and so forth down the line until the requested information is retrieved or no more web sites left to check.

The web site ranking method and system of the present invention provides robustness to the voice browser system and enables it to adapt to changes that may occur as web sites evolve. For instance, the information required by a web site **114** to perform a search or the format of the reported response data may change. Without the ability to adequately monitor and detect these changes, a search requested by a user may provide an incomplete response, no response, or an error. Such useless responses may result from incomplete data being provided to the web site **114** or the web browsing server **102** being unable to recognize the response data messages received from the searched web site **114**.

The robustness and reliability of the voice browsing system of the present invention is further improved by the addition of a polling mechanism. This polling mechanism continually polls or "pings" each of the sites listed in the database **100**. During this polling function, a web browsing server **102** sends brief data requests or "polling digital data" to each web site listed in database **100**. The web browsing server **102** monitors the response received from each web site and determines whether it is a complete response and whether the response is in the expected format specified by the content descriptor file **406** used by the extraction agent **400**. The polled web sites that provide complete responses in the format expected by the extraction agent **400** have their ranking established based on their "response time". That is, web sites with faster response times will be will be assigned higher rankings than those with slower response times If the web browsing server **102** receives no response from the polled web site or if the response received is not in the expected format, then the rank of that web site is lowered. Additionally, the web browsing server contains a warning

mechanism that generates a warning message or alarm for the system administrator indicating that the specified web site has been modified or is not responsive and requires further review.

Since the web browsing servers **102** access web sites based upon their ranking number, only those web sites that produce useful and error-free responses will be used by the voice browser system to gather information requested by the user. Further, since the ranking numbers are also based upon the speed of a web site in providing responses, only the most time efficient sites are accessed. This system assures that users will get complete, timely, and relevant responses to their requests. Without this feature, users may be provided with information that is not relevant to their request or may not get any information at all. The constant polling and re-ranking of the web sites used within each category allows the voice browser of the present invention to operate efficiently. Finally, it allows the voice browser system of the present invention to dynamically adapt to changes in the rapidly evolving web sites that exist on the Internet.

It should be noted that the web sites accessible by the voice browser of the preferred embodiment may use any type of mark-up language, including Extensible Markup Language (XML), Wireless Markup Language (WML), Handheld Device Markup Language (HDML), Hyper Text Markup Language (HTML), or any variation of these languages.

A second embodiment of the present invention is depicted in FIG. **5**. This embodiment provides a system and method for controlling a variety of devices **500** connected to a network **502** by using conversational speech commands spoken into a voice enabled device **504** (i.e., wireline or wireless telephones, Internet Protocol (IP) phones, or other special wireless units). The networked devices may include various household devices. For instance, voice commands may be used to control household security systems, VCRs, TVs, outdoor or indoor lighting, sprinklers, or heating and air conditioning systems.

Each of these devices **500** is connected to a network **502**. These devices **500** may contain embedded microprocessors or may be connected to other computer equipment that allow the device **500** to communicate with network **502**. In the preferred embodiment, the devices **500** appear as "web sites" connected to the network **502**. This allows a network interface system, such as a device browsing server **506**, a database **508**, and a user interface system, such as a media server **510**, to operate similar to the web browsing server **102**, database **100** and media server **106** described in the first preferred embodiment above. A network **502** interfaces with one or more network interface systems, which are shown as device browsing servers **506** in FIG. **5**. The device browsing servers perform many of the same functions and operate in much the same way as the web browsing servers **102** discuss above in the first preferred embodiment. The device browsing servers **506** are also connected to a database **508**.

Database **508** lists all devices that are connected to the network **502** For each device **500**, the database **508** contains a record similar to that shown in FIG. **2** Each record will contain at least a device identifier, which may be in the form of a URL, and a command to "content extraction agent" contained in the device browsing server **506**. Database **508** may also include any other data or software necessary to test and administer the device browsing system.

The content extraction agent operates similarly to that described in the first embodiment. A device descriptor file contains a listing of the options and functions available for each of the devices **500** connected on the network **502**. Furthermore, the device descriptor file contains the information necessary to properly communicate with the networked devices **500**. Such information would include, for example, communication protocols, message formatting requirements, and required operating parameters.

The device browsing server **506** receives messages from the various networked devices **500**, appropriately formats those messages and transmits them to one or more media servers **510** which are part of the device browsing system. The user's voice enabled devices **504** can access the device browsing system by calling into a media server **510** via the Public Switched Telephone Network (PSTN) **512**. In the preferred embodiment, the device browsing server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The media servers **510** act as user interface systems and perform the functions of natural speech recognition, speech synthesis, data processing, and call handling. The media server **510** operates similarly to the media server **106** depicted in FIG. **3**. When data is received from the device browser server **506**, the media server **510** will convert the data into audio messages via a speech synthesis engine that are then transmitted to the voice enabled device of the user **504**. Speech commands received from the voice enabled device of the user **504** are converted into data messages via a speech recognition engine running on the media server **510**. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www nuance.com). A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhsl.com). The media servers **510** of the preferred embodiment are based on Intel's Dual Pentium III 730 MHz microprocessor system. A specific example for using the system and method of this embodiment of the invention will now be given.

First, a user may call into a media server **510** by dialing a telephone number associated with an established device browsing system. Once the user is connected, the IUR application of the media server **510** will provide the user with a list of available systems that may be monitored or controlled based upon information contained in database **508**.

For example, the user may be provided with the option to select "Home Systems" or "Office Systems". The user may then speak the command "access home systems". The media server **510** would then access the database **508** and provide the user with a listing of the home subsystems or devices **500** available on the network **502** for the user to monitor and control. For instance, the user may be given a listing of subsystems such as "Outdoor Lighting System", "Indoor Lighting System", "Security System", or "Heating and Air Conditioning System". The user may then select the indoor lighting subsystem by speaking the command "Indoor Lighting System". The IUR application would then provide the user with a set of options related to the indoor lighting system. For instance the media server **510** may then provide a listing such as "Dining Room", "Living Room", "Kitchen", or "Bedroom". After selecting the desired room, the IUR application would provide the user with the options to hear the "status" of the lighting in that room or to "turn on", "turn off", or "dim" the lighting in the desired room. These commands are provided by the user by speaking the desired command into the users voice enabled device **504**. The media server **510** receives this command and translates it into a data message. This data message is then forwarded

to the device browsing server **506** which routes the message to the appropriate device **500**.

The device browsing system **514** of this embodiment of the present invention also provides the same robustness and reliability features described in the first embodiment. The device browsing system **514** has the ability to detect whether new devices have been added to the system or whether current devices are out-of-service. This robustness is achieved by periodically polling or "pinging" all devices **500** listed in database **508**. The device browsing server **506** periodically polls each device **500** and monitors the response. If the device browsing server **506** receives a recognized and expected response from the polled device, then the device is categorized as being recognized and in-service. However, if the device browsing server **506** does not receive a response from the polled device **500** or receives an unexpected response, then the device **500** is marked as being either new or out-of-service. A warning message or a report may then be generated for the user indicating that a new device has been detected or that an existing device is experiencing trouble.

Therefore, this embodiment allows users to remotely monitor and control any devices that are connected to a network, such as devices within a home or office. Furthermore, no special telecommunications equipment is required for users to remotely access the device browser system. Users may use any type of voice enabled device (i.e., wireline or wireless telephones, IP phones, or other wireless units) available to them. Furthermore, a user may perform these functions from anywhere without having to subscribe to additional services. Therefore, no additional expenses are incurred by the user.

The descriptions of the preferred embodiments described above are set forth for illustrative purposes and are not intended to limit the present invention in any manner. Equivalent approaches are intended to be included within the scope of the present invention. While the present invention has been described with reference to the particular embodiments illustrated, those skilled in the art will recognize that many changes and variations may be made thereto without departing from the spirit and scope of the present invention. These embodiments and obvious variations thereof are contemplated as falling within the scope and spirit of the claimed invention.

What is claimed is:

1. An Internet voice browsing system for gathering information from web sites on the Internet, comprising:

a CPU-based media server, said media server including at least a speech recognition engine, a speech synthesis engine, an interactive voice response application, a call processing system, and telephony hardware, said media server configured to receive a speech command from a user and to convert said speech command into a digital data message;

a database containing a list of web sites stored on magnetic media,

a rank number assigned to each one of said web sites and stored in said database;

a CPU-based web browsing server, said web browsing server including at least a content extraction agent, a content fetcher, a polling and ranking agent, and a content descriptor file, said web browsing server configured to receive said digital data message from said media server and configured to access one of said web sites having the highest said rank number and to retrieve information from said one of said web sites;

said media server configured to generate an audio message representing said information and to transmit said audio message to said user; and

a polling mechanism configured to periodically send a polling digital data message to each one of said web sites and to receive a response, each of said web sites thereby becoming a polled web site, said polling mechanism configured to decrease said rank number of said polled web site if no response is received from said polled web site, said polling mechanism also configured to decrease said rank number of said polled web site if an unexpected response is received from said polled web site, and said polling mechanism also configured to decreases said rank number of said polled web site if a response time of said polled web site is longer than a second response time of a second polled web site.

2. A method for using voice commands to browse Internet web sites, comprising the steps of:

providing a database storing a list of web sites on magnetic media;

assigning a rank number to each of said web sites and storing said rank number in said database;

receiving a voice command from a user and converting said command into a digital data message;

providing a CPU-based web browsing system for receiving said digital data message and accessing one of said web sites having the highest said rank number, said web browsing system including at least a content extraction agent, a content fetcher, a polling and ranking agent, and a content file;

receiving at said web browsing system response data from said web site with the highest rank number;

converting said response data into an audio message that is transmitted to said user;

periodically polling each of said web sites listed in said database, each of said web sites thereby becoming a polled web site;

decreasing said rank number of said polled web site if no response is received from said polled web site;

decreasing said rank number of said polled web site if an unexpected response is received from said polled web site; and

decreasing said rank number of said polled web site if a response time of said polled web site is longer than a second response time of a second polled web site.

3. A system for remotely controlling household devices, including security systems, lighting systems, heating and air conditioning systems, TVs, and VCRs, comprising:

a CPU-based media server, said media server including at least a speech recognition engine, a speech synthesis engine, an interactive voice response application, a call processing system, and telephony hardware, said media server configured to receive a speech command from a user;

at least one household device connected to a network;

a CPU-based device browsing server, connected with said media server and said network, said device browsing server including at least a content extraction agent, a content fetcher, a polling agent, and a content descriptor file, said device browsing agent configured to access at least one of said household devices in response to said speech command and configured to control the operation of said one of said household devices,

a polling mechanism configured to periodically send a polling digital data message to each said household

device and to receive a response, said polling mechanism configured to evaluate said response from a polled household device, and

a warning mechanism configured to generate a warning message if said response data provided by said polled household device cannot be recognized by said polling mechanism.

4. A method for remotely controlling household devices, including security systems, lighting systems, heating and air conditioning systems, TVs, or VCRs, comprising the steps of:

receiving at a user interface system a speech command from a user;

providing at least one household device connected to a network,

accessing by a CPU-based network interface system at least one said household device in response to said speech command in order to control the operation of said one household device;

polling each said household device and attempting to recognize response data received from each polled household device; and

generating a warning message if said response data provided by said polled household device cannot be recognized.

* * * * *

Exhibit 2

US007516190B2

(12) **United States Patent**
Kurganov

(10) **Patent No.: US 7,516,190 B2**
(45) **Date of Patent: Apr. 7, 2009**

(54) **PERSONAL VOICE-BASED INFORMATION RETRIEVAL SYSTEM**

(75) Inventor: **Alexander Kurganov**, Buffalo Grove, IL (US)

(73) Assignee: **Parus Holdings, Inc.**, Bannockburn, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 458 days.

(21) Appl. No.: **09/777,406**

(22) Filed: **Feb. 6, 2001**

(65) **Prior Publication Data**

US 2001/0054085 A1 Dec. 20, 2001

Related U.S. Application Data

(60) Provisional application No. 60/180,343, filed on Feb. 4, 2000.

(51) **Int. Cl.**
G06F 15/16 (2006.01)
G10L 21/00 (2006.01)

(52) **U.S. Cl.** **709/217**; 709/203; 704/270.1; 704/275

(58) **Field of Classification Search** ..... 709/218, 709/219, 317, 224, 203; 379/105, 67, 88; 370/354, 79; 179/2 A, 18; 364/188, 514; 455/417; 704/275, 270.1
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,313,035	A	1/1982	Jordan et al.	179/18
4,585,906	A	4/1986	Matthews et al.	179/18
4,596,900	A	6/1986	Jackson	179/2 A
4,696,028	A	9/1987	Morganstein et al.	379/88
4,761,807	A	8/1988	Matthews et al.	379/89
4,776,016	A	10/1988	Hansen	381/42
4,809,321	A	2/1989	Morganstein et al.	379/211
4,837,798	A	6/1989	Cohen et al.	379/88
4,850,012	A	7/1989	Mehta et al.	379/157
4,873,719	A	10/1989	Reese	379/215
4,907,079	A	3/1990	Turner et al.	358/84
4,922,526	A	5/1990	Morganstein et al.	379/157

(Continued)

FOREIGN PATENT DOCUMENTS

GB	2317782	1/1998
WO	96/09710	3/1996
WO	WO 97/37481	10/1997
WO	98/23058	5/1998

OTHER PUBLICATIONS

Ly, "Chatter A: A Conversational Telephone Agent," *submitted to Program in Media Arts and Sciences*, Massachusetts Institute of Technology, (1993), pp. 1-130.

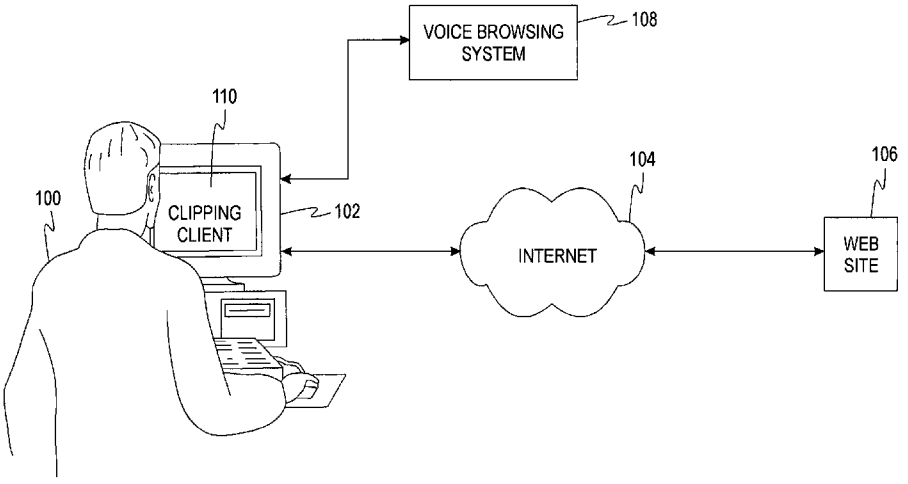
(Continued)

Primary Examiner—William C Vaughn, Jr.
Assistant Examiner—Kristie D Shingles
(74) *Attorney, Agent, or Firm*—Foley & Lardner LLP

(57) **ABSTRACT**

The present invention relates to a system for retrieving information from a network such as the Internet. A user creates a user-defined record in a database that identifies an information source, such as a web site, containing information of interest to the user. This record identifies the location of the information source and also contains a recognition grammar based upon a speech command assigned by the user. Upon receiving the speech command from the user that is described within the recognition grammar, a network interface system accesses the information source and retrieves the information requested by the user.

21 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

4,933,966	A		6/1990	Hird et al. 379/132
4,935,958	A		6/1990	Morganstein et al. 379/372
4,953,204	A		8/1990	Cuschleg, Jr. et al. 379/266
4,955,047	A		9/1990	Morganstein et al. 379/112
4,972,462	A		11/1990	Shibata 379/89
4,975,941	A		12/1990	Morganstein et al. 379/88
4,994,926	A		2/1991	Gordon et al. 358/400
5,020,095	A		5/1991	Morganstein et al. 379/67
5,027,384	A		6/1991	Morganstein et al. 379/67
5,029,196	A		7/1991	Morganstein 379/67
5,086,385	A		2/1992	Launey et al. 364/188
5,099,509	A		3/1992	Morganstein et al. 379/84
5,109,405	A		4/1992	Morganstein 379/89
5,131,024	A		7/1992	Pugh et al. 379/67
5,166,974	A		11/1992	Morganstein et al. 379/67
5,195,086	A		3/1993	Baumgartner et al. 370/264
5,243,645	A		9/1993	Bissell et al. 379/211
5,249,219	A		9/1993	Morganstein et al. 379/84
5,263,084	A		11/1993	Chaput et al. 379/215
5,291,302	A		3/1994	Gordon et al. 358/400
5,291,479	A		3/1994	Vaziri
5,303,298	A		4/1994	Morganstein et al. 379/67
5,307,399	A		4/1994	Dai
5,309,504	A		5/1994	Morganstein et al. 379/67
5,325,421	A		6/1994	Hou et al. 379/67
5,327,486	A		7/1994	Wolff et al. 379/96
5,327,529	A		7/1994	Fults et al. 395/155
5,329,578	A		7/1994	Brennan et al. 379/67
5,333,266	A		7/1994	Boaz et al. 709/206
5,347,574	A		9/1994	Morganstein et al. 379/210
5,355,403	A		10/1994	Richardson, Jr. et al. 379/88
5,375,161	A		12/1994	Fuller et al. 379/57
5,384,771	A		1/1995	Isidoro et al. 370/58.2
5,404,231	A		4/1995	Bloomfield 358/400
5,408,526	A		4/1995	McFarland et al. 379/202
5,414,754	A		5/1995	Pugh et al. 379/67
5,436,963	A		7/1995	Fitzpatrick et al. 379/212
5,459,584	A		10/1995	Gordon et al. 358/434
5,463,684	A		10/1995	Morduch et al. 379/202
5,475,791	A		12/1995	Schalk et al. 395/2.42
5,497,373	A		3/1996	Hulen et al. 370/79
5,499,288	A		3/1996	Hunt et al. 379/88
5,517,558	A		5/1996	Schalk 379/88
5,555,100	A		9/1996	Bloomfield 358/402
5,559,611	A		9/1996	Bloomfield et al. 358/407
5,559,859	A		9/1996	Dai
5,566,236	A		10/1996	MeLampy et al. 379/201
5,603,031	A		2/1997	White et al. 709/317
5,608,786	A		3/1997	Gordon 379/100
5,610,970	A		3/1997	Fuller et al. 455/417
5,611,031	A		3/1997	Hertzfeld et al. 345/433
5,652,789	A		7/1997	Miner et al. 379/201
5,657,376	A		8/1997	Espeut et al.
5,659,597	A		8/1997	Bareis 455/563
5,666,401	A		9/1997	Morganstein et al. 379/67
5,675,507	A		10/1997	Bobo, II ..... 364/514 R
5,675,811	A		10/1997	Broedner et al. 395/750
5,689,669	A		11/1997	Lynch et al. 345/355
5,692,187	A		11/1997	Goldman et al. 395/619
5,699,486	A	*	12/1997	Tullis et al. 704/270.1
5,719,921	A		2/1998	Vystosky et al. 379/88
5,721,908	A	*	2/1998	Lagarde et al. 707/10
5,724,408	A		3/1998	Morganstein et al. 379/88
5,742,905	A		4/1998	Pepe et al. 455/461
5,752,191	A		5/1998	Fuller et al. 455/445
5,787,298	A		7/1998	Broedner et al. 395/750
5,793,993	A		8/1998	Broedner et al. 395/306
5,799,065	A		8/1998	Junqua et al.
5,809,282	A		9/1998	Cooper et al. 359/500
5,812,796	A		9/1998	Broedner et al. 395/283
5,819,220	A	*	10/1998	Sarukkai et al. 704/270.1
5,819,306	A		10/1998	Goldman et al. 711/100
5,867,494	A		2/1999	Krishnaswamy et al. 370/352
5,867,495	A		2/1999	Elliott et al. 370/352
5,873,080	A		2/1999	Coden et al. 707/3
5,881,134	A		3/1999	Foster et al. 379/88
5,884,262	A		3/1999	Wise et al. 704/270
5,884,266	A	*	3/1999	Dvorak 704/270.1
5,890,123	A	*	3/1999	Brown et al. 704/270.1
5,915,001	A	*	6/1999	Uppaluru 379/88.22
5,953,392	A	*	9/1999	Rhie et al. 379/88.13
5,974,413	A		10/1999	Beauregard et al. 707/6
5,999,525	A		12/1999	Krishnaswamy et al. 370/352
6,012,088	A		1/2000	Li et al. 709/219
6,014,437	A		1/2000	Acker et al. 379/219
6,018,710	A		1/2000	Wynblatt et al. 704/260
6,021,181	A		2/2000	Miner et al. 379/88.23
6,031,904	A		2/2000	An et al. 379/201
6,038,305	A		3/2000	McAllister et al. 379/207
6,047,053	A		4/2000	Miner et al. 379/201
6,078,580	A		6/2000	Mandalia et al. 370/352
6,081,518	A		6/2000	Bowman-Amuah 370/352
6,091,808	A		7/2000	Wood et al. 379/201
6,101,472	A	*	8/2000	Giangarra et al. 704/275
6,104,803	A		8/2000	Weser et al. 379/230
6,115,742	A		9/2000	Franklin et al. 709/224
6,208,638	B1		3/2001	Rieley et al. 370/354
6,233,318	B1		5/2001	Picard et al. 379/88.17
6,243,373	B1		6/2001	Turock 370/352
6,252,944	B1		6/2001	Hansen, II et al. 379/67.1
6,269,336	B1	*	7/2001	Ladd et al. 704/270
6,285,745	B1		9/2001	Bartholomew et al. ... 379/88.17
6,327,572	B1	*	12/2001	Morton et al. 705/10
6,349,132	B1	*	2/2002	Wesemann et al. 379/88.17
6,353,661	B1	*	3/2002	Bailey, III 379/93.25
6,366,578	B1		4/2002	Johnson 370/353
6,434,529	B1	*	8/2002	Walker et al. 704/275
6,446,076	B1	*	9/2002	Burkey et al. 707/102
6,456,699	B1	*	9/2002	Burg et al. 379/88.17
6,477,420	B1		11/2002	Struble et al. 379/67.1
6,505,163	B1		1/2003	Zhang et al. 704/275
6,529,948	B1	*	3/2003	Bowman-Amuah 709/217
6,532,444	B1	*	3/2003	Weber 704/257
6,539,359	B1	*	3/2003	Ladd et al. 704/275
6,546,393	B1		4/2003	Khan 707/10
6,594,348	B1	*	7/2003	Bjurstrom et al. 379/88.13
6,618,726	B1	*	9/2003	Colbath et al. 707/6
6,636,831	B1	*	10/2003	Profit et al. 704/275
6,665,640	B1	*	12/2003	Bennett et al. 704/257
6,687,341	B1	*	2/2004	Koch et al. 379/88.17
6,718,015	B1	*	4/2004	Berstis 379/88.17
6,732,142	B1	*	5/2004	Bates et al. 709/203
6,771,743	B1	*	8/2004	Butler et al. 379/67.1
6,823,370	B1	*	11/2004	Kredo et al. 709/206
6,888,929	B1	*	5/2005	Saylor et al. 379/88.16
6,922,733	B1	*	7/2005	Kuiken et al. 709/246
6,941,273	B1	*	9/2005	Loghmani et al. 705/26
6,964,012	B1	*	11/2005	Zirngibl et al. 715/513
6,965,864	B1	*	11/2005	Thrift et al. 704/275
6,996,609	B2	*	2/2006	Hickman et al. 709/218
7,050,977	B1	*	5/2006	Bennett 704/270.1
2001/0032234	A1	*	10/2001	Summers et al. 709/201
2001/0048676	A1	*	12/2001	Jimenez et al. 370/352
2002/0006126	A1	*	1/2002	Johnson et al. 370/356

OTHER PUBLICATIONS

Schmandt et al., "Phone Slave: A Graphical Telecommunications Interface," *Proceedings of the SID* (1985), vol. 26/1, pp. 79-82.

Don Hunt and Brian Edwards "*Long-Distance Remote Control to the Rescue*." Chicago Tribune, Jun. 15, 2002, Section 4, p. 15.

* cited by examiner

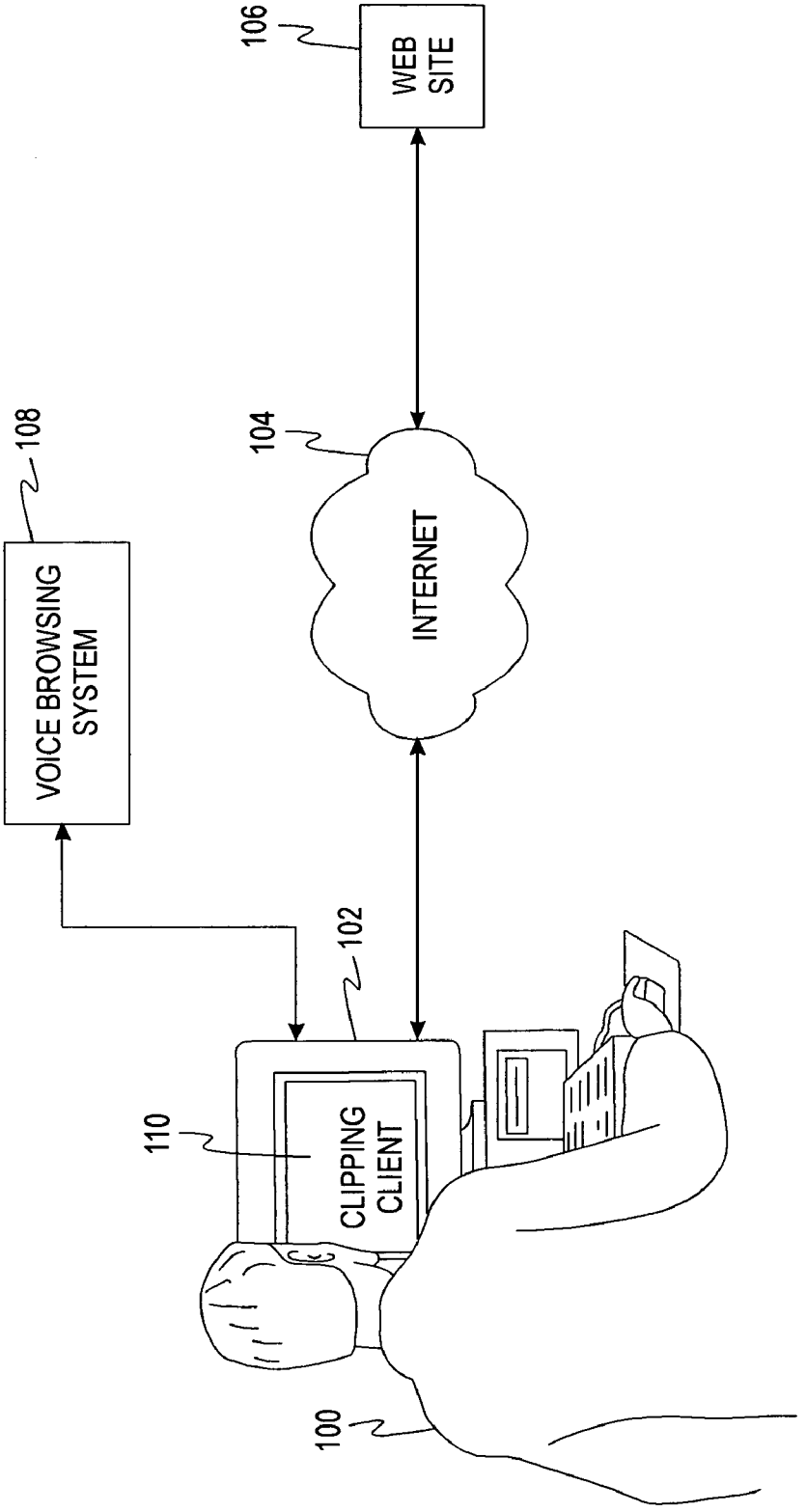

106
WEB SITE
104
INTERNET
108
VOICE BROWSING SYSTEM
102
110
CLIPPING CLIENT
100

FIG. 1

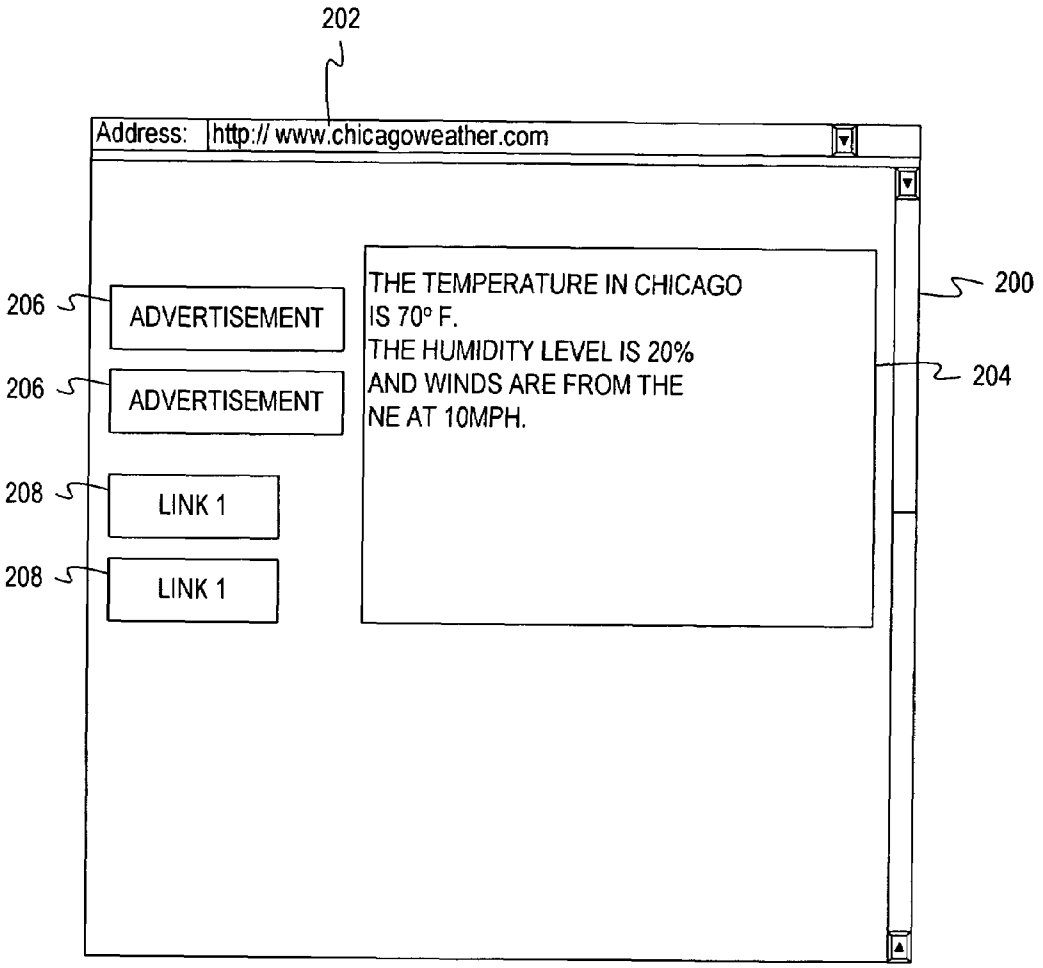

202
Address: http:// www.chicagoweather.com
206
ADVERTISEMENT
206
ADVERTISEMENT
208
LINK 1
208
LINK 1
THE TEMPERATURE IN CHICAGO
IS 70° F.
THE HUMIDITY LEVEL IS 20%
AND WINDS ARE FROM THE
NE AT 10MPH.
200
204

FIG. 2

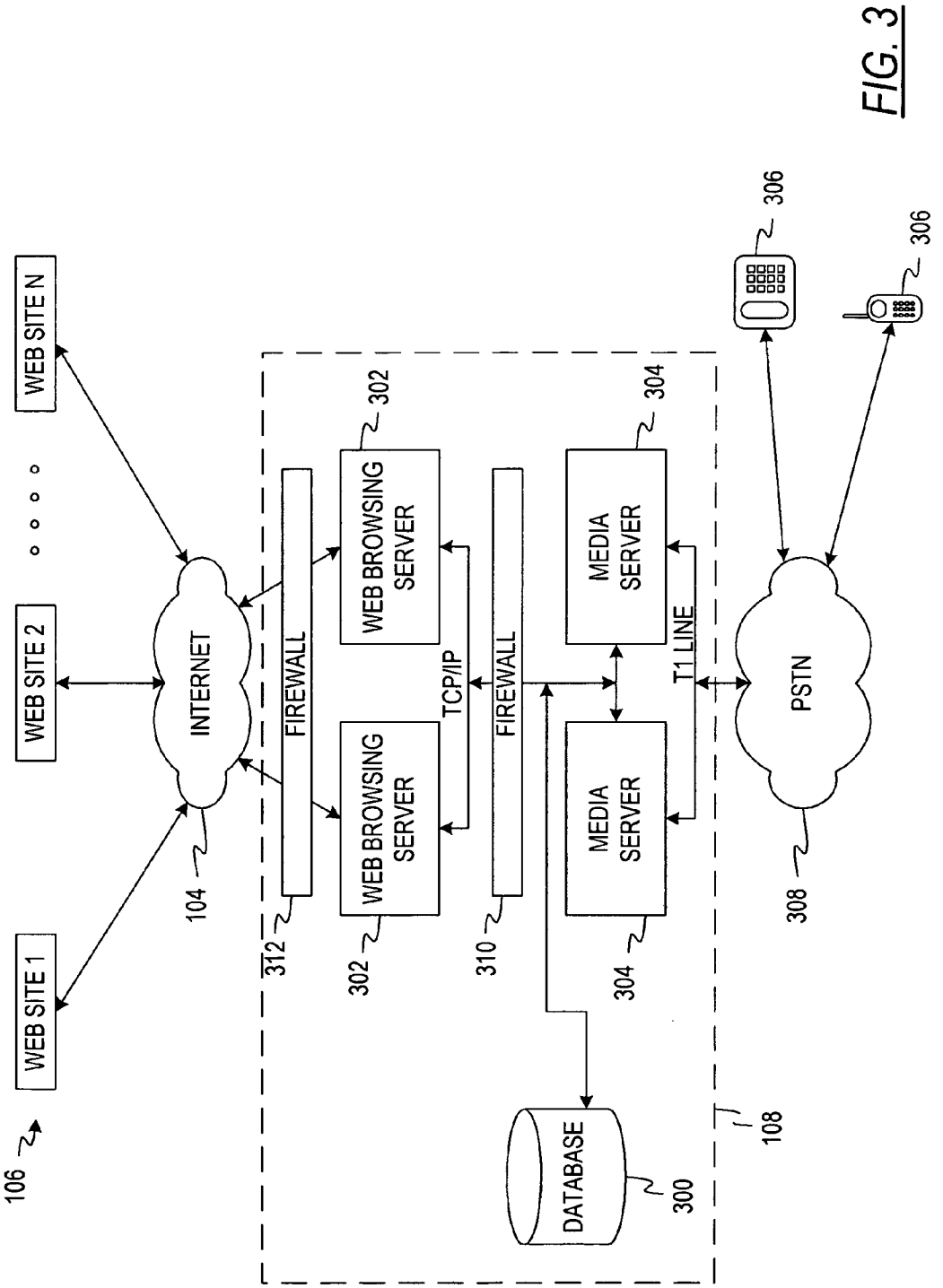

WEB SITE N
WEB SITE 2
WEB SITE 1
106
INTERNET
104
FIREWALL
312
WEB BROWSING SERVER
302
WEB BROWSING SERVER
302
TCP/IP
FIREWALL
310
MEDIA SERVER
304
MEDIA SERVER
304
T1 LINE
PSTN
308
306
306
DATABASE
300
108

FIG. 3

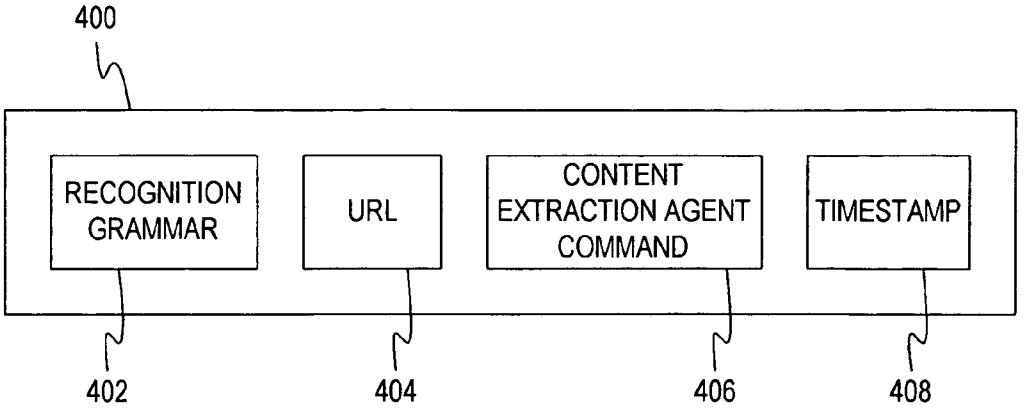

400
RECOGNITION GRAMMAR
URL
CONTENT EXTRACTION AGENT COMMAND
TIMESTAMP
402
404
406
408

FIG. 4

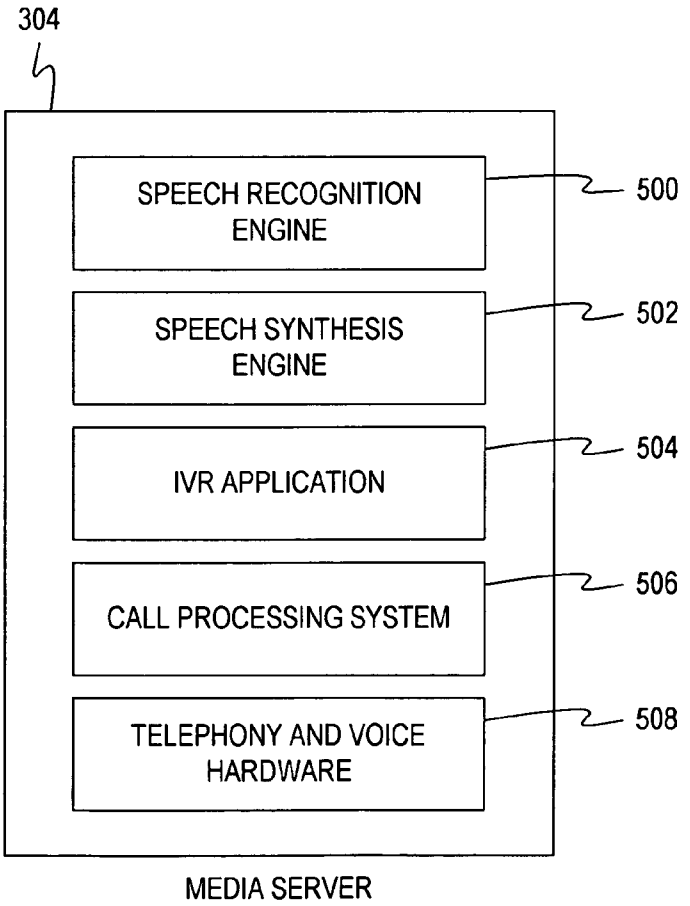

304
SPEECH RECOGNITION ENGINE
500
SPEECH SYNTHESIS ENGINE
502
IVR APPLICATION
504
CALL PROCESSING SYSTEM
506
TELEPHONY AND VOICE HARDWARE
508
MEDIA SERVER

FIG. 5

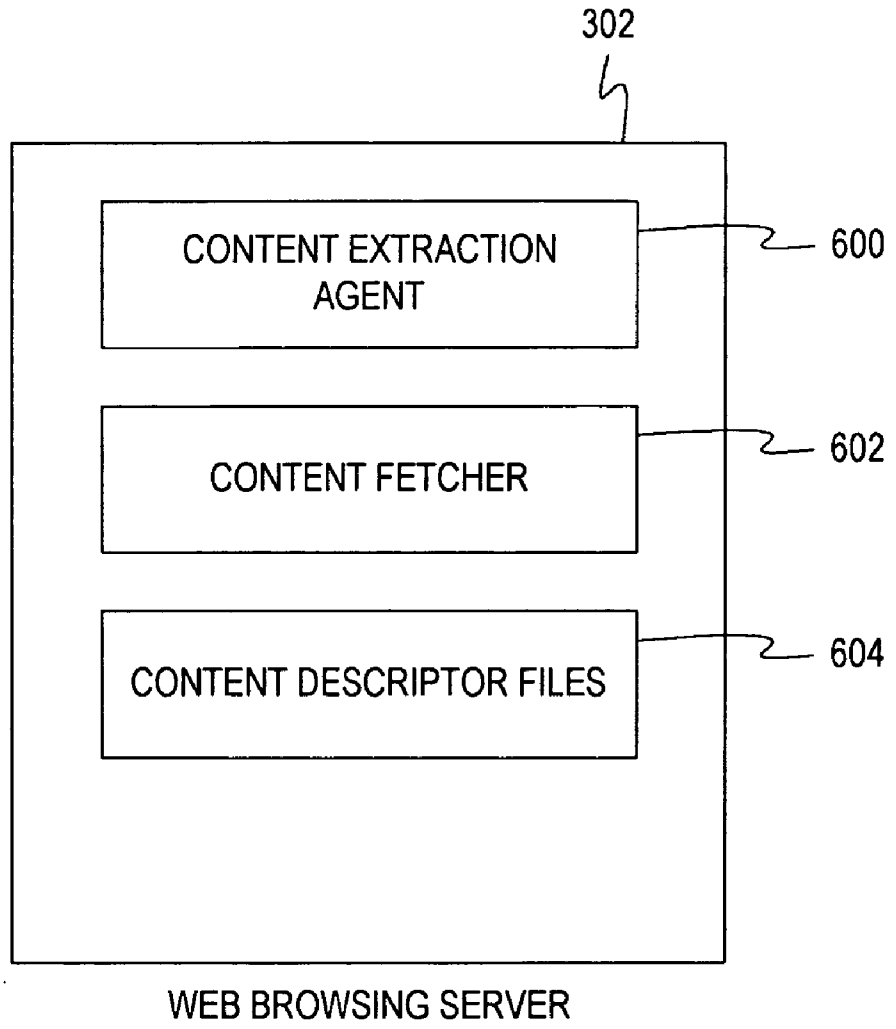

302
CONTENT EXTRACTION AGENT
600
CONTENT FETCHER
602
CONTENT DESCRIPTOR FILES
604
WEB BROWSING SERVER

FIG. 6

PERSONAL VOICE-BASED INFORMATION RETRIEVAL SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to U.S. Provisional Application Ser. No. 60/180,343, filed Feb. 4, 2000 entitled "Personal Voice-Based Information Retrieval System."

FIELD OF THE INVENTION

The present invention relates generally to the field of providing information access. In particular, the invention relates to a personalized system for accessing information from the Internet or other information sources using speech commands.

BACKGROUND OF THE INVENTION

Popular methods of information access and retrieval using the Internet or other computer networks can be time-consuming and complicated. A user must frequently wade through vast amounts of information provided by an information source or web site in order obtain a small amount of relevant information. This can be time-consuming, frustrating, and, depending on the access method, costly. A user is required to continuously identify reliable sources of information and, if these information sources are used frequently, repeatedly access these sources.

Current methods of accessing information stored on computer networks, such as Wide Area Networks (WANs), Local Area Network (LANs) or the Internet, require a user to have access to a computer. While computers are becoming increasingly smaller and easier to transport, using a computer to access information is still more difficult than simply using a telephone. Since speech recognition systems allow a user to convert his voice into a computer-usable message, telephone access to digital information is becoming more and more feasible. Voice recognition technology is growing in its ability to allow users to use a wide vocabulary. Further, such technology is quite accurate when a single, known user only needs to use a small vocabulary.

Therefore, a need exists for an information access and retrieval system and method that allows users to access frequently needed information from information sources on networks by using a telephone and simple speech commands.

SUMMARY OF THE INVENTION

One object of the preferred embodiment of the present invention is to allow users to customize a voice browsing system.

A further object of the preferred embodiment is to allow users to customize the information retrieved from the Internet or other computer networks and accessed by speech commands over telephones.

Another object of the preferred embodiment is to provide a secure and reliable retrieval of information over the Internet or other computer networks using predefined verbal commands assigned by a user.

The present invention provides a solution to these and other problems by providing a new system for retrieving information from a network such as the Internet. A user creates a user-defined record in a database that identifies an information source, such as a web site, containing information of interest to the user. This record identifies the location of the information source and also contains a recognition grammar assigned by the user. Upon receiving a speech command from the user that is described in the assigned recognition grammar, a network interface system accesses the information source and retrieves the information requested by the user.

In accordance with the preferred embodiment of the present invention, a customized, voice-activated information access system is provided. A user creates a descriptor file defining specific information found on a web site the user would like to access in the future. The user then assigns a pronounceable name or identifier to the selected content and this pronounceable name is saved in a user-defined database record as a recognition grammar along with the URL of the selected web site.

In the preferred embodiment, when a user wishes to retrieve the previously defined web-based information, a telephone call is placed to a media server. The user provides speech commands to the media server that are described in the recognition grammar assigned to the desired search. Based upon the recognition grammar, the media server retrieves the user-defined record from a database and passes the information to a web browsing server which retrieves the information from associated web site. The retrieved information is then transmitted to the user using a speech synthesis software engine.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** displays a personal information selection system used with the preferred embodiment of the present invention;

FIG. **2** displays a web page displayed by the clipping client of the preferred embodiment,

FIG. **3** is a block diagram of a voice browsing system used with preferred embodiment of the present invention;

FIG. **4** is a block diagram of a user-defined database record created by preferred embodiment of the present invention;

FIG. **5** is a block diagram of a media server used by the preferred embodiment, and

FIG. **6** is a block diagram of a web browsing server used by the preferred embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention uses various forms of signal and data transmission to allow a user to retrieve customized information from a network using speech communication. In the preferred embodiment of the present invention, a user associates information of interest found on a specific information source, such as a web site, with a pronounceable name or identification word. This pronounceable name/identification word forms a recognition grammar in the preferred embodiment. When the user wishes to retrieve the selected information, he may use a telephone or other voice enabled device to access a voice browser system. The user then speaks a command described in the recognition grammar associated with the desired information. The voice browsing system then accesses the associated information source and returns to the user, using a voice synthesizer, the requested information.

Referring to FIG. **1**, a user **100** uses a computer **102** to access a network, such as a WAN, LAN, or the Internet, containing various information sources. In the preferred embodiment, the user **100** access the Internet **104** and begins searching for web sites **106**, which are information sources that contain information of interest to the user. When the user **100** identifies a web site **106** containing information the user would like to access using only a voice enabled device, such

as a telephone, and the voice browsing system **108**, the user initiates a "clipping client" engine **110** on his computer **102**.

The clipping client **110** allows a user **100** to create a set of instructions for use by the voice browsing system **108** in order to report personalized information back to the user upon request. The instruction set is created by "clipping" information from the identified web site A user **100** may be interested in weather for a specific city, such as Chicago. The user **100** identifies a web site from which he would like to obtain the latest Chicago weather information. The clipping client **110** is then activated by the user **100**.

The clipping client **110** displays the selected web site in the same manner as a conventional web browser such as Microsoft's® Internet Explorer. FIG. **2** depicts a sample of a web page **200** displayed by the clipping client **110**. The user **100** begins creation of the instruction set for retrieving information from the identified web site by selecting the uniform resource locator (URL) address **202** for the web site (i.e., the web site address). In the preferred embodiment, this selection is done by highlighting and copying the URL address **202**. Next, the user selects the information from the displayed web page that he would like to have retrieved when a request is made. Referring to FIG. **2**, the user would select the information regarding the weather conditions in Chicago **204**. The web page **200** may also contain additional information such as advertisements **206** or links to other web sites **208** which are not of interest to the user. The clipping client **110** allows the user to select only that portion of the web page containing information of interest to the user. Therefore, unless the advertisements **206** and links **208** displayed on the web page are of interest to the user, he would not select this information. Based on the web page information **204** selected by the user, the clipping client **110** creates a content descriptor file containing a description of the content of the selected web page. This content descriptor file indicates where the information selected by the user is located on the web page. In the preferred embodiment, the content descriptor file is stored within the web browsing server **302** shown in FIG. **3**. The web browsing server **302** will be discussed below.

Table 1 below is an example of a content descriptor file created by the clipping client of the preferred embodiment. This content descriptor file relates to obtaining weather information from the web site www.cnn.com.

TABLE 1

```
table name : portalServices
column :
      service
content:
            weather
column:
      config
content:
      [cnn]
      Input=__zip
      URL=http://cgi.cnn.com/cgi-bin/weather/redirect?zip=__zip
      Pre-filter="\n"    "
      Pre-filter="<[^<>]+>" "
      Pre-filter=/\s+/ /
      Pre-filter=" [\(\)\|]" ! "
      Output=__location
      Output=first__day__name
      Output=first__day__weather
      Output=first__day__high__F
      Output=first__day__high__C
      Output=first__day__low__F
      Output=first__day__low__C
      Output=second__day__name
      Output=second__day__weather
      Output=second__day__high__F
```

TABLE 1-continued

```
      Output=second__day__high__C
      Output=second__day__low__F
      Output=second__day__low__C
      Output=third__day__name
      Output=third__day__weather
      Output=third__day__high__F
      Output=third__day__high__C
      Output=third__day__low__F
      Output=third__day__low__C
      Output=fourth__day__name
      Output=fourth__day__weather
      Output=fourth__day__high__F
      Output=fourth__day__high__C
      Output=fourth__day__low__F
      Output=fourth__day__low__C
      Output=undef
      Output=__current__time
      Output=__current__month
      Output=__current__day
      Output=__current__weather
      Output=__current__temperature__F
      Output=__current__temperature__C
      Output=__humidity
      Output=__wind
      Output=__pressure
      Output=__sunrise
      Output=__sunset
      Regular__expression=WEB SERVICES: (.+) Forecast FOUR-DAY
FORECAST (\S+)
(\S+) HI
GH (\S+) F (\S+) C LOW (\S+) F (\S+) C (\S+) (\S+) HIGH
(\S+) F (\S+) C LOW
(\S+
) F (\S+) C (\S+) (\S+) HIGH (\S+) F (\S+) C LOW (\S+) F
(\S+) C (\S+) (\S+)
HIG
H (\S+) F (\S+) C LOW (\S+) F (\S+) C WEATHER MAPS RADAR
(.+) Forecast
CURRENT C
ONDITIONS (.+) !local!, (\S+) (\S+) (.+) Temp: (\S+) F,
(\S+) C Rel.
Humidity: (
\S+) Wind: (.+) Pressure: (.+) Sunrise: (.+) Sunset: (.+)
```

Finally, the clipping client **110** prompts the user to enter an identification word or phrase that will be associated with the identified web site and information. For example, the user could associate the phrase "Chicago weather" with the selected URL **202** and related weather information **204**. The identification word or phrase is stored as a personal recognition grammar that can now be recognized by a speech recognition engine of the voice browsing system **108** which will be discussed below. The personal recognition grammar, URL address **202**, and a command for executing a content extraction agent are stored within a database used by the voice browser system **108** which will be discussed below.

The voice browsing system **108** used with the preferred embodiment will now be described in relation to FIG. **3**. A database **300** designed by Webley Systems Incorporated is connected to one or more web browsing servers **302** as well as to one or more media servers **304**. The database may store information on magnetic media, such as a hard disk drive, or it may store information via other widely acceptable methods for storing data, such as optical disks. The media servers **304** function as user interface systems that provide access to the voice browsing system **108** from a user's voice enabled device **306** (i.e, any type of wireline or wireless telephone, Internet Protocol (IP) phones, or other special wireless units). The database **300** contains a section that stores the personal recognition grammars and related web site information generated by the clipping client **110**. A separate record exists for each web site defined by the user. An example of a user-

defined web site record is shown in FIG. **4**. Each user-defined web site record **400** contains the recognition grammar **402** assigned by the user, the associated Uniform Resource Locator CURL) **404**, and a command that enables the "content extraction agent" **406** and retrieves the appropriate content descriptor file required to generate proper requests to the web site and to properly format received data. The web-site record **400** also contains the timestamp **408** indicating the last time the web site was accessed. The content exaction agent is described in more detail below.

The database **300** may also contain a listing of pre-recorded audio files used to create concatenated phrases and sentences. Further, database **300** may contain customer profile information, system activity reports, and any other data or software servers necessary for the testing or administration of the voice browsing system **108**.

The operation of the media servers **304** will now be discussed in relation to FIG. **5**. The media servers **304** function as user interface systems since they allow a user to access the voice browsing system **108** via a voice enabled device **306**. In the preferred embodiment, the media servers **304** contain a speech recognition engine **500**, a speech synthesis engine **502**, an Interactive Voice Response (IVR) application **504**, a call processing system **506**, and telephony and voice hardware **508** that is required to enable the voice browsing system **108** to communicate with the Public Switched Telephone Network (PSTN) **308**. In the preferred embodiment, each media server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The speech recognition function is performed by a speech recognition engine **500** that converts voice commands received from the user's voice enabled device **306** (i.e., any type of wireline or wireless telephone, Internet Protocol (IP) phones, or other special wireless units) into data messages. In the preferred embodiment, voice commands and audio messages are transmitted using the PSTN **308** and data is transmitted using the TCP/IP communications protocol. However, one skilled in the art would recognize that other transmission protocols may be used. Other possible transmission protocols would include SIP/VoIP (Session Initiation Protocol/Voice over IP), Asynchronous Transfer Mode (ATM) and Frame Relay. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www.nuance.com). The Nuance engine capacity is measured in recognition units based on CPU type as defined in the vendor specification. The natural speech recognition grammars (i e., what a user can say that will be recognized by the speech recognition grammars (i,e., what a user can say that will be recognized by the speech recognition engine) were developed by Webley Systems.

In the preferred embodiment, when a user access the voice browsing system **108**, he will be prompted if he would like to use his "user-defined searches." If the user answers affirmatively, the media servers **304** will retrieve from the database **300** the personal recognition grammars **402** defined by the user while using the clipping client **110**.

The media servers **304** also contain a speech synthesis engine **502** that converts the data retrieved by the web browsing servers **302** into audio messages that are transmitted to the user's voice enabled device **306**. A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhsl.com)

A further description of the web browsing server **302** will be provided in relation to FIG. **6**. The web browsing servers **302** provide access to data stored on any computer network including the Internet **104**, WANs or LANs. The web browsing servers **302** receive responses from web sites **106** and extract the data requested by the user. This task is known as "content extraction." The web browsing server **302** is comprised of a content extraction agent **600**, a content fetcher **602**, and the content descriptor file **604**. Each of these are software applications and will be discussed below.

Upon receiving a user-defined web site record **400** from the database **300** in response to a user request, the web browsing server **302** invokes the "content extraction agent" command **406** contained in the record **400**. The content extraction agent **600** retrieves the content descriptor file **604** associated with the user-defined record **400**. As mentioned, the content descriptor file **604** directs the extraction agent where to extract data from the accessed web page and how to format a response to the user utilizing that data. For example, the content descriptor file **604** for a web page providing weather information would indicate where to insert the "city" name or ZIP code in order to retrieve Chicago weather information. Additionally, the content descriptor file **604** for each supported URL indicates the location on the web page where the response information is provided. The extraction agent **600** uses this information to properly extract from the web page the information requested by the user.

The content extraction agent **600** can also parse the content of a web page in which the user-desired information has changed location or format. This is accomplished based on the characteristic that most hypertext documents include named objects like tables, buttons, and forms that contain textual content of interest to a user. When changes to a web page occur, a named object may be moved within a document, but it still exists. Therefore, the content extraction agent **600** simply searches for the relevant name of desired object. In this way, the information requested by the user may still be found and reported regardless of changes that have occurred.

Table 2 below contains source code for a content extraction agent **600** used by the preferred embodiment.

TABLE 2

```
#!/usr/local/www/bin/sybper15
#$Header:
/usr/local/cvsroot/webley/agents/service/web_dispatch.pl,v
1.6
# Dispatches all web requests
#http://wcorp.itn.net/cgi/flstat?carrier=ua&flight_no=155&m
on_abbr=jul&date=
6&stamp=OhLN~PdbuuE*itn/ord,itn/cb/sprint_hd
#http://cgi.cnnfn.com/flightview/rlm?airline=amt&number=300
require "config_tmp.pl";
# check parameters
die "Usage: $0 service [params]\n" if $#ARGV < 1;
#print STDERR @ARGV;
# get parameters
my ( $service, @param ) = @ARGV;
# check service
my %Services = (
                    weather_cnn => 'webget.pl weather_cnn',
                    weather_lycos => 'webget.pl
weather_lycos',
                    weather_weather => 'webget.pl
weather_weather',
                    weather_snap => 'webget.pl
weather_snap',
                    weather_infospace => 'webget.pl
weather_infospace',
                    stockQuote_yahoo => 'webget.pl stock',
                    flightStatus_itn => 'webget.pl
flight_delay',
                    yellowPages_yahoo => 'yp_data.pl',
                    yellowPages_yahoo => 'yp_data.pl',
                    newsHeaders_newsreal => 'news.pl',
                    newsArticle_newsreal => 'news.pl',
                    );
```

TABLE 2-continued

```
# test param
my $date = `date`;
chop ( $date );
my ( $short_date ) = $date =~ /\s+(\w{3}\s+\d{1,2})\s+/;
my %Test = (
                    weather_cnn => '60053',
                    weather_lycos => '60053',
                    weather_weather => '60053',
                    weather_snap => '60053',
                    weather_infospace => '60053',
                    stockQuote_yahoo => 'msft',
                    flightStatus_itn => 'ua 155 ' .
$short_date,
                    yellowPages_yahoo => 'tires 60015',
                    newsHeaders_newsreal => '1',
                    newsArticle_newsreal => '1  1',
                    );
die "$date: $0: error: no such service: $service (check
this script)\n"
unless $Services{ $service };
# prepare absolute path to run other scripts
my ( $path, $script ) = $0 =~ m|^(.*/) ([^/]*) |;
# store the service to compare against datatable
my $service_stored = $service;
# run service
while( ! ( $response = `$path$Services{ $service } @param` )
)   {
        # response failed
        # check with test parameters
        $response = `$path$Services{ $service } $Test{
$service }`;
#       print "test: $path$Services{ $service } $Test{
$service }";
        if ( $response )    {
            $service = &switch_service( $service );
#           print "Wrong parameter values were supplied:
$service -
@param\n";
#           die "$date: $0: error: wrong parameters: $service
-
@param\n";
        }
        else {
            # change priority and notify
            $service = &increase_attempt( $service );
        }
}
# output the response
print $response;
sub increase_attempt {
        my ( $service ) = @_;
        my ( $service_name ) = split( /_/, $service );
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                    . "set priority = ( select max( priority
) from
mcServiceRoute "
                    . "where service = '$service_name' ) + 1,
"
                    . "date = getdate( ), "
                    . "attempt = attempt + 1 "
                    . "where route = '$script $service' " );
#       print "---$route===\n";
        # find new route
        my $route = @{ &db_query( "select route from
mcServiceRoute "
                                . "where service =
'$service_name'  "
                                . "and attempt < 5
"
                                . "order by
priority ")
                        } -> [ 0 ] { route };
        &db_query( "update mcServiceRoute "
                    . "set attempt = 0 "
                    . "where route = '$script $service' " )
```

TABLE 2-continued

```
            if ( $route eq "$script $service"
                    or $route eq "$script $service_stored" );
        ( $service_name, $service ) = split( /\s+/, $route );
        die "$date: $0: error: no route for the service:
$service (add
more) \n"
            unless $service;
        return $service;
}
sub switch_service {
        my ( $service ) = @_;
        my ( $service_name ) = split( /_/, $service );
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                    . "set priority = ( select max( priority
) from
mcServiceRoute "
                    . "where service = '$service_name' ) + 1,
"
                    . "date = getdate( ) "
                    . "where route = '$script $service' " );
#       print "---$route===\n";
        # find new route
        my $route = @{ &db_query( "select route from
mcServiceRoute "
                                . "where service =
'$service_name'  "
                                . "and attempt < 5
"
                                . "order by
priority ")
                        } -> [ 0 ] { route };
        die "$date: $0: error: there is the only service:
$route (add
more)\n"
            if ( $route eq "$script $service"
                    or $route eq "$script $service_stored" );
        ( $service_name, $service ) = split( /\s+/, $route );
        die "$date: $0: error: no route for the service:
$service (add
more)\n"
            unless $service;
        return $service;
}
```

Table 3 below contains source code of the content fetcher **602** used with the content extraction agent **600** to retrieve information from a web site

TABLE 3

```
#!/usr/local/www/bin/sybper15
#-T
# -w
# $Header:
/usr/local/cvsroot/webley/agents/service/webget.pl,v 1.4
# Agent to get info from the web.
# Parameters: service_name [service_parameters], i.e. stock
msft or weather
60645
# Configuration stored in files service_name.ini
# if this file is absent the configuration is received from
mcServices table
# This script provides autoupdate to datatable if the .ini
file is newer.
$debug = 1;
use URI::URL;
use LWP::UserAgent;
use HTTP::Request::Common;
use Vail::VarList;
use Sybase::CTlib;
use HTTP::Cookies;
#print "Sybase::CTlib $DB_USR, $DB_PWD, $DB_SRV;";
open( STDERR, ">>$0.log" ) if $debug;
```

TABLE 3-continued

```
#open( STDERR, ">&STDOUT" );
$log = `date`;
#$response = `./url.pl
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"`;
#$response = `pwd`;
#print STDERR "pwd = $response\n";
#$response = `ls`;
#print STDERR "ls = $response\n";
chop( $log );
$log .= "pwd=" . `pwd`;
chop( $log );
#$debug2 = 1;
my $service = shift;
$log .= " $service: ". join( ':', @ARGV ) . "\n";
print STDERR $log if $debug;
#$response = `./url.pl
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"`;
my @ini = &read__ini( $service );
chop( @ini );
my $section = " ";
do { $section = &process__section( $section ) } while
$section;
#$response = `./url.pl
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"`;
exit;
################################################
sub read__ini {
        my ( $service ) = @__;
        my @ini = ( );
        # first, try to read file
        $0 =~ m|^(.*/) [^/]*|;
        $service = $1 . $service;
        if ( open( INI, "$service.ini" ) ) {
                @ini = ( <INI> );
                return @ini unless ( $DB__SRV );
                # update datatable
                my $file__time = time - int( ( -M "$service.ini" )
* 24 *
3600 );
#               print "time $file__time\n";
                my $dbh = new Sybase::CTlib $DB__USR, $DB__PWD,
$DB__SRV;
                unless ( $dbh ) {
                        print STDERR "webget.pl: Cannot connect to
dataserver $DB__SRV:$DB__USR:$DB__PWD\n";
                        return @ini;
                }
                my @row__refs = $dbh->ct__sql( "select lastUpdate
from
mcServices where service = '$service' ", undef, 1 );
                if ( $dbh->{ RC } == CS__FAIL ) {
                        print STDERR "webget.pl: DB select from
mcServices
failed\n";
                        return @ini;
                }
                unless ( defined @row__refs ) {
                        # have to insert
                        my ( @ini__escaped ) = map {
                                ( my $x = $__ ) =~ s/\'/\'\'/g;
                                $x;
                        } @ini;
                        $dbh->ct__sql( "insert mcServices values(
'$service',
'@ini__escaped', $file__time )" );
                        if ( $dbh->{ RC } == CS__FAIL ) {
                                print STDERR "webget.pl: DB insert to
mcServices failed\n";
                        }
                        return @ini;
                }
#               print "time $file__time:".$row__refs[ 0 ]->{
'lastUpdate'
}."\n";
                if ( $file__time > $row__refs[ 0 ]->{ 'lastUpdate'
} ) {
                        # have to update
                        my ( @ini__escaped ) = map {
                                ( my $x = $__ ) =~ s/\'/\'\'/g;
                                $x;
                        } @ini;
                        $dbh->ct__sql( "update mcServices set config
=
'@ini__escaped', lastUpdate = $file__time where service =
'$service' " );
                        if ( $dbh->{ RC } == CS__FAIL ) {
                                print STDERR "webget.pl: DB update to
mcServices failed\n";
                        }
                }
                return @ini;
        }
        else {
                print STDERR "$0: WARNING: $service.ini n/a in "
. `pwd`
                        . "Try to read DB\n";
        }
        # then try to read datatable
        die "webget.pl: Unable to find service $service\n"
unless ( $DB__SRV
);
        my $dbh = new Sybase::CTlib $DB__USR, $DB__PWD,
        $DB__SRV;
        die "webget.pl: Cannot connect to dataserver
$DB__SRV:$DB__USR:$DB__PWD\n" unless ( $dbh );
        my @row__refs = $dbh->ct__sql( "select config from
mcServices where
service = '$service' ", undef, 1 );
        die "webget.pl: DB select from mcServices failed\n" if
$dbh->{ RC }
== CS__FAIL;
        die "webget.pl: Unable to find service $service\n"
unless ( defined
@row__refs );
        $row__refs[ 0 ]->{ 'config' } =~ s/\n /\n\r/g;
        @ini = split( /\r/, $row__refs[ 0 ]->{ 'config' } );
        return @ini;
}
################################################
sub process__section {
        my ( $prev__section ) = @__;
        my ( $section, $output, $content );
        my %Param;
        my %Content;
#       print"############################\n";
        foreach ( @ini ) {
#               print;
#               chop;
                s/\s+$//;
                s/^\s+//;
                # get section name
                if ( /^\[(.*)\]/ ) {
#                       print "$__: $section:$prev__section\n";
                        last if $section;
                        next if $1 eq "print";
#                       next if $prev__section ne " " and
$prev__section ne $1;
                        if ( $prev__section eq $1 ) {
                                $prev__section = " ";
                                next;
                        }
                        $section = $1;
                }
                # get parameters
                push( @{ $Param{ $1 } }, $2 ) if $section and
/([^=]+)=(.*)/;
        }
#       print"++++++++++++++++++++++++++++++\n";
        return 0 unless $section;
#       print "section $section\n";
        # substitute parameters with values
        map { $Param{ URL }->[ 0 ] =~ s/$Param{ Input }->[ $__
]/$ARGV[ $__
]/g
        } 0 . . $#{ $Param{ Input } };
        # get page content
        ( $Content{ 'TIME' }, $content ) = &get__url__content(
```

TABLE 3-continued

```
${ $Param{ URL
} } [ 0 ] );
        # filter it
        map {
            if ( /\"([^\"]+)\"([^\"]*)\"/ or
/\/([^\/]+)\/([^\/]*)\// )
{
                    my $out = $2; $content =~ s/$1/$out/g;
            }
        } @{ $Param{ "Pre-filter" } };
#print STDERR $content;
        # do main regular expression
        unless( @values = $content =~ /${ $Param{
Regular_expression } } [ 0
]/ ) {
            &die_hard( ${ $Param{ Regular_expression } } [ 0
], $content
);
            return $section;
        }
        %Content = map { ( $Param{ Output }->[ $_ ], $values[
$_ ] )
        } 0 . . $#{ $Param{ Output } };
        # filter it
        map {
            if ( /([^\"]+)\"([^\"]+)\"([^\"]*)\"/
                or /([^\/]+)\/([^\/]+)\/([^\/]*)\// ) {
                  my $out = $3;
                  $Content{ $1 } =~ s/$2/$out/g;
            }
        } @{ $Param{ "Post-filter" } };
        # calculate it
        map {
            if ( /([^=]+)=(.*)/ ) {
                my $eval = $2;
                map { $eval =~ s/$_/$Content{ $_ }/g
                } keys %Content;
               $Content{ $1 } = eval( $eval );
            }
        } @{ $Param{ Calculate } };
        # read section [print]
        foreach $i ( 0 . . $#ini ) {
            next unless $ini[ $i ] =~ /^\[print\]/;
            foreach ( $i + 1 . . $#ini ) {
                last if $ini[ $_ ] =~ /^\[.+\]/;
                $output .= $ini[ $_ ] . "\n";
            }
            last;
        }
        # prepare output
        map { $output =~ s/$_/$Content{ $_ }/g
        } keys %Content;
        print $output;
        return 0;
}
##################################################
sub get_url_content {
        my ( $url ) = @_;
        print STDERR $url if $debug;
#       $response = `./url.pl '$url'`;
        $response = `./url.pl '$url'`;
        return( $time - time, $response );
        my $ua = LWP::UserAgent->new;
        $ua->agent( 'Mozilla/4.0 [en] (X11; I; FreeBSD 2.2.8-
STABLE i386) '
);
#       $ua->proxy( ['http', 'https'],
'http://proxy.webley:3128/' );
#       $ua->no_proxy( 'webley', 'vail' );
        my $cookie = HTTP::Cookies->new;
        $ua->cookie_jar( $cookie );
        $url = url $url;
        print "$url\n" if $debug2;
        my $time = time;
        my $res = $ua->request( GET $url );
        print "Response: " . ( time - $time ) . "sec\n" if
$debug2;
        return( $time - time, $res->content );
}
```

TABLE 3-continued

```
##################################################
sub die_hard {
        my( $re, $content ) = @_;
        my ( $re_end, $pattern );
        while( $content !~ /$re/ ) {
            if ( $re =~ s/(\([^\(\)]+\) [^\(\)]*$)// ) {
                $re_end = $1 . $re_end;
            }
            else {
                $re_end = $re;
                last;
            }
        }
        $content =~ /$re/;
        print STDERR "The regular expression did not match:\n
$re\n
Possible misuse:
$re_end:\n
Matched:
$&\n
Mismatched:
$'\n
" if $debug;
            if ( $debug ) {
                print STDERR "Content:\n $content\n" unless
$';
            }
}
##################################################
```

Once the web browsing server **302** accesses the web site specified in the URL **404** and retrieves the requested information, it is forwarded to the media server **304**. The media server uses the speech synthesis engine **502** to create an audio message that is then transmitted to the user's voice enabled device **306**. In the preferred embodiment, each web browsing server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

Referring to FIG. **3**, the operation of the personal voice-based information retrieval system will be described. A user establishes a connection between his voice enabled device **306** and a media server **304** of the voice browsing system **108**. This may be done using the Public Switched Telephone Network (PSTN) **308** by calling a telephone number associated with the voice browsing system **108**. Once the connection is established, the media server **304** initiates an interactive voice response (IVR) application. The IVR application plays audio message to the user presenting a list of options, which includes "perform a user-defined search." The user selects the option to perform a user-defined search by speaking the name of the option into the voice enabled device **306**.

The media server **304** then accesses the database **300** and retrieves the personal recognition grammars **402**. Using the speech synthesis engine **502**, the media server **304** then asks the user, "Which of the following user-defined searches would you like to perform" and reads to the user the identification name, provided by the recognition grammar **402**, of each user-defined search. The user selects the desired search by speaking the appropriate speech command or pronounceable name described within the recognition grammar **402**. These speech recognition grammars **402** define the speech commands or pronounceable names spoken by a user in order to perform a user-defined search. If the user has a multitude of user-defined searches, he may speak the command or pronounceable name described in the recognition grammar **402** associated with the desired search at anytime without waiting for the media server **304** to list all available user-defined searches. This feature is commonly referred to as a "barge-in" feature.

The media server **304** uses the speech recognition engine **500** to interpret the speech commands received from the user. Based upon these commands, the media server **304** retrieves the appropriate user-defined web site record **400** from the database **300**. This record is then transmitted to a web browsing server **302**. A firewall **310** may be provided that separates the web browsing server **302** from the database **300** and media server **304**. The firewall provides protection to the media server and database by preventing unauthorized access in the event the firewall **312** for the web browsing server fails or is compromised. Any type of firewall protection technique commonly known to one skilled in the art could be used, including packet filter, proxy server, application gateway, or circuit-level gateway techniques.

The web browsing server **302** accesses the web site **106** specified by the URL **404** in the user-defined web site record **400** and retrieves the user-defined information from that site using the content extraction agent and specified content descriptor file specified in the content extraction agent command **406**. Since the web browsing server **302** uses the URL and retrieves new information from the Internet each time a request is made, the requested information is always updated.

The content information received from the responding web site **106** is then processed by the web browsing server **302** according to the associated content descriptor file. This processed response is then transmitted to the media server **304** for conversion into audio messages using either the speech synthesis engine **502** or selecting among a database of prerecorded voice responses contained within the database **300**.

It should be noted that the web sites accessible by the personal information retrieval system and voice browser of the preferred embodiment may use any type of mark-up language, including Extensible Markup Language (XML), Wireless Markup Language (WML), Handheld Device Markup Language (HDML), Hyper Text Markup Language (HTML), or any variation of these languages.

The descriptions of the preferred embodiments described above are set forth for illustrative purposes and are not intended to limit the present invention in any manner Equivalent approaches are intended to be included within the scope of the present invention. While the present invention has been described with reference to the particular embodiments illustrated, those skilled in the art will recognize that many changes and variations may be made thereto without departing from the spirit and scope of the present invention. These embodiments and obvious variations thereof are contemplated as falling within the scope and spirit of the claimed invention.

I claim:

1. A method for allowing users to use speech commands to obtain information from a pre-defined portion of a pre-selected web site in audio format, said method comprising the steps of:

(a) providing a computer having a speech processor, said computer being operatively connected to the internet and to at least one phone;

(b) providing a URL to said computer, said URL indicating a pre-selected web site from which the information is to be retrieved;

(c) using said computer to designate a pre-defined portion of the pre-selected web site which contains the information to be retrieved;

(d) using said computer to identify a named object associated with the content of the information to be retrieved;

(e) using said computer to generate a regular expression based on said pre-defined portion of said pre-selected web site and said named object, said regular expression corresponding to said content of said information to be retrieved, wherein said regular expression is a text string used for describing a search pattern;

(f) providing a speech command to said speech processor, said speech command corresponding to said regular expression;

(g) said speech processor converting said speech command to a digital-form command;

(h) said computer receiving said digital-form command from said speech processor, said computer assigning said regular expression to said digital-form command;

(i) after steps (a) through (h) are completed, transmitting an audio speech command to said speech processor, said speech command corresponding to said regular expression;

(j) said speech processor converting said speech command to said digital-form command;

(k) said computer receiving said digital-form command from said speech processor;

(l) said computer retrieving said regular expression corresponding to said digital-form command;

(m) said computer retrieving the information from the pre-defined portion of the pre-selected web site corresponding to said regular expression when the requested information is found in the pre-defined portion of the pre-selected website;

(n) said computer searching said pre-selected web site for said named object when the requested information is not found in the pre-defined portion of the pre-selected web site;

(o) said computer providing said retrieved information to said speech processor;

(p) said speech processor converting said retrieved information into an audio message; and

(q) said speech processor forwarding said audio message to a user.

2. The method of claim **1** wherein the pre-defined portion of the pre-selected web site being retrieved is periodically updated.

3. The method of claim **1** wherein the step of providing a URL to a computer is performed by a user.

4. The method of claim **1** wherein the step of using said computer to designate a pre-defined portion of the web site which contains the information to be retrieved comprises the steps of:

displaying the web site on a graphical display operatively connected to the computer; and

using computer software to select the pre-defined portion of the pre-selected web site which contains the information to be retrieved.

5. The method of claim **4** wherein the step of using said computer to designate a pre-defined portion of the web site which contains the information to be retrieved is performed by a user.

6. A system for retrieving information from a pre-defined portion of a pre-selected web site by uttering speech commands into a phone and for providing to a user retrieved information in an audio form, said system comprising:

a server, said server operatively connected to the internet and to at least one phone, said server comprising:

telephony hardware, said telephony hardware operatively connected to said phone and to said server;

at least one speech recognition engine, said speech recognition engine operatively connected to said server and to said telephony hardware;

a speech synthesis engine, said speech synthesis engine operatively connected to said server and to said telephony hardware; and

a call processing system, said call processing system configured to receive speech commands through said telephony hardware and forward said speech commands to said speech recognition engine and said call processing system further configured to receive an audio message from said speech synthesis engine and forward said audio message through said telephony hardware;

a first instruction set stored on said server, said first instruction set configured to identify the pre-defined portion of the pre-selected web site and to identify a named object associated with the content of the information to be retrieved, said pre-defined portion containing the information to be retrieved from the web site, said first instruction set comprising:

a uniform resource locator address for said web site; and

the named object;

a second instruction set stored on said server, said second instruction set configured to generate a regular expression based on said pre-defined portion of said pre-selected web site and said named object, said regular expression corresponding to said content of said information to be retrieved, wherein said regular expression is a text string used for describing a search pattern;

a recognition grammar corresponding to each said instruction set and corresponding to a speech command;

said speech recognition engine configured to receive said speech command and to select the corresponding recognition grammar, said speech recognition engine further configured to retrieve each said instruction set corresponding to said recognition grammar upon receiving said speech command;

a web browser operatively connected to said server, said web browser including at least a content extraction agent, a content fetcher, and a content descriptor file, said web browser configured to access said pre-defined portion of said web she defined by said instruction sets and to retrieve said information defined by said instruction sets;

said speech synthesis engine configured to convert the retrieved information from said pre-defined portion of said pre-selected web site into an audio message, and said speech synthesis engine further configured to transmit said audio message to said user.

7. The system of claim **6** wherein the phone is a landline telephone.

8. The system of claim **6** wherein the phone is a wireless telephone.

9. The system of claim **6** wherein the phone is an internet protocol telephone.

10. The system of claim **6** wherein the server is operatively connected to a local area network.

11. The system of claim **6** wherein the server is operatively connected to a wide area network.

12. The system of claim **6** wherein the server is operatively connected to the Internet.

13. The system of claim **6** further comprising a database operatively connected to the server, the database configured to store said instruction set and said recognition grammars.

14. The system of claim **6** further comprising computer software stored on the server, said computer software configured to create said instruction set based on user-defined information.

15. The system of claim **6** further comprising:

a graphical display operatively connected to the server, said graphical display configured to display the pre-selected web site; and

computer software stored on the server, said computer software configured to select the pre-defined portion of the pre-selected web site which contains the information to be retrieved.

16. A method for allowing a phone user to set up and subsequently retrieve information in an audio format from a pre-defined portion of a pre-selected web she, said method comprising the steps of:

providing a server operatively connected to the internet and to at least one phone, said server being operatively connected to a speech recognition engine and to a speech synthesis engine;

providing a first instruction set stored on said server for identifying the pre-defined portion of a pre-selected web she containing the content of the information to be retrieved from the web site, said first instruction set comprising:

a uniform resource locater address for said web site; and

a named object associated with the content of the information to be retrieved;

providing a second instruction set stored on said server for generating a regular expression based on said pre-defined portion of said pre-selected web site and said named object, said regular expression corresponding to said content of said information to be retrieved, wherein said regular expression is a text string used for describing a search pattern;

providing a speech command to said speech recognition engine, said speech command corresponding to said instruction sets;

said speech recognition engine assigning said speech command to a recognition grammar, said speech command and said recognition grammar corresponding to each said instruction set;

transmitting said speech command to said speech recognition engine;

said speech recognition engine receiving said speech command and selecting the corresponding recognition grammar;

said server retrieving each said instruction set corresponding to said recognition grammar;

said server accessing said pre-defined portion of said pre-selected web site defined by said instruction sets and retrieving said information defined by said instruction sets when the requested information is found in the pre-defined portion of the pre-selected web site;

said server searching said pre-selected website when the requested information is not found in the pre-defined portion of the pre-selected web site;

said speech synthesis engine converting the retrieved information from said pre-selected web site into an audio message; and

said speech synthesis engine transmitting said audio message to said user.

17. The method of claim **16** wherein the step of providing at least one instruction set to the server is performed by the user.

18. The method of claim **16** wherein the step of providing at least one instruction set to the server comprises the steps of:

displaying the web site on a graphical display operatively connected to the server; and

using computer software to select the pre-defined portion of the pre-selected web site which contains the information to be retrieved.

19. The method of claim **18** wherein the step of providing at least one instruction set to the server is performed by the user.

20. The method of claim **16** wherein the pre-defined portion of the pre-selected web site being retrieved is periodically updated.

21. The system of claim **6** wherein the named object is selected from the group consisting of: "weather", "forecast", "high", "low", "radar", "temp", "temperature", "humidity", "humidity level", "wind", "wind speed", "wind direction", "pressure", "sunrise", "sunset", "time", "month", "day", "stock", "stock quote", "news", "news reel", "airline", "carrier", "flight", and "flight number".

* * * * *

EXHIBIT 3

US008185402B2

(12) **United States Patent**
Kurganov et al.

(10) **Patent No.: US 8,185,402 B2**
(45) **Date of Patent: *May 22, 2012**

(54) **ROBUST VOICE BROWSER SYSTEM AND VOICE ACTIVATED DEVICE CONTROLLER**

(75) Inventors: **Alexander Kurganov**, Buffalo Grove, IL (US); **Valery Zhukoff**, Deerfield, IL (US)

(73) Assignee: **Parus Holdings, Inc.**, Bannockburn, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/973,475**

(22) Filed: **Dec. 20, 2010**

(65) **Prior Publication Data**

US 2011/0091023 A1 Apr. 21, 2011

Related U.S. Application Data

(63) Continuation of application No. 12/030,556, filed on Feb. 13, 2008, now Pat. No. 7,881,941, which is a continuation of application No. 11/409,703, filed on Apr. 24, 2006, now Pat. No. 7,386,455, which is a continuation of application No. 10/821,690, filed on Apr. 9, 2004, now Pat. No. 7,076,431, which is a continuation of application No. 09/776,996, filed on Feb. 5, 2001, now Pat. No. 6,721,705.

(60) Provisional application No. 60/180,344, filed on Feb. 4, 2000, provisional application No. 60/233,068, filed on Sep. 15, 2000.

(51) **Int. Cl.**
G10L 21/06 (2006.01)

(52) **U.S. Cl.** **704/275**; 379/88.17

(58) **Field of Classification Search** 704/275; 379/88.17

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,728,486	A	4/1973	Kraus
4,058,838	A	11/1977	Crager et al.
4,100,377	A	7/1978	Flanagan
4,313,035	A	1/1982	Jordan et al.
4,327,251	A	4/1982	Fomenko et al.
4,340,783	A	7/1982	Sugiyama et al.
4,371,752	A	2/1983	Matthews et al.
4,481,574	A	11/1984	DeFino et al.
4,489,438	A	12/1984	Hughes

(Continued)

FOREIGN PATENT DOCUMENTS

CA	1329852	5/1994

(Continued)

OTHER PUBLICATIONS

"A PABX that Listens and Talks", Speech Technology, Jan./Feb. 1984, pp. 74-79.

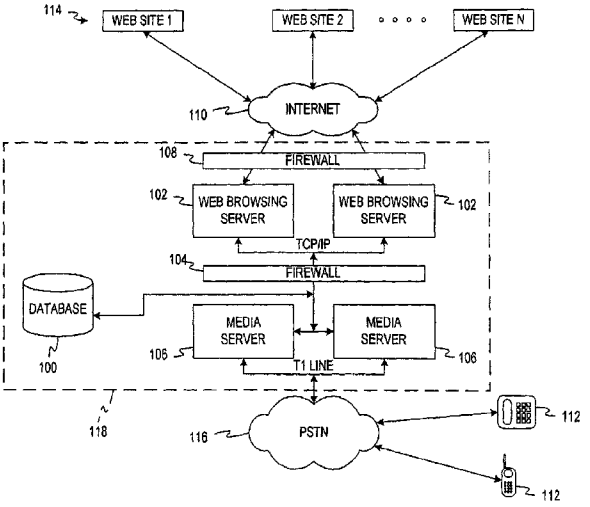
(Continued)

Primary Examiner — Susan McFadden
(74) *Attorney, Agent, or Firm* — Berry & Associates P.C.

(57) **ABSTRACT**

The present invention relates to a system for acquiring information from sources on a network, such as the Internet. A voice browsing system maintains a database containing a list of information sources, such as web sites, connected to a network. Each of the information sources is assigned a rank number which is listed in the database along with the record for the information source. In response to a speech command received from a user, a network interface system accesses the information source with the highest rank number in order to retrieve information requested by the user.

15 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

4,500,751	A	2/1985	Darland et al.
4,513,390	A	4/1985	Walter et al.
4,523,055	A	6/1985	Hohl et al.
4,549,047	A	10/1985	Brian et al.
4,584,434	A	4/1986	Hashimoto
4,585,906	A	4/1986	Matthews et al.
4,596,900	A	6/1986	Jackson
4,602,129	A	7/1986	Matthews et al.
4,635,253	A	1/1987	Urui et al.
4,652,700	A	3/1987	Matthews et al.
4,696,028	A	9/1987	Morganstein et al.
4,713,837	A	12/1987	Gordon
4,747,127	A	5/1988	Hansen et al.
4,748,656	A	5/1988	Gibbs et al.
4,755,932	A	7/1988	Diedrich
4,757,525	A	7/1988	Matthews et al.
4,761,807	A	8/1988	Matthews et al.
4,763,317	A	8/1988	Lehman et al.
4,769,719	A	9/1988	Endo
4,771,425	A	9/1988	Baran et al.
4,776,016	A	10/1988	Hansen
4,782,517	A	11/1988	Bernardis et al.
4,792,968	A	12/1988	Katz
4,799,144	A	1/1989	Parruck et al.
4,809,321	A	2/1989	Morganstein et al.
4,811,381	A	3/1989	Woo et al.
4,837,798	A	6/1989	Cohen et al.
4,847,891	A	7/1989	Kotani
4,850,012	A	7/1989	Mehta et al.
4,852,149	A	7/1989	Zwick et al.
4,866,758	A	9/1989	Heinzelmann
4,873,719	A	10/1989	Reese
4,879,743	A	11/1989	Burke et al.
4,893,333	A	1/1990	Baran et al.
4,893,335	A	1/1990	Fuller et al.
4,903,289	A	2/1990	Hashimoto
4,905,273	A	2/1990	Gordon et al.
4,907,079	A	3/1990	Turner et al.
4,918,722	A	4/1990	Duehren et al.
4,922,518	A	5/1990	Gordon et al.
4,922,520	A	5/1990	Bernard et al.
4,922,526	A	5/1990	Morganstein et al.
4,926,462	A	5/1990	Ladd et al.
4,930,150	A	5/1990	Katz
4,933,966	A	6/1990	Hird et al.
4,935,955	A	6/1990	Neudorfer
4,935,958	A	6/1990	Morganstein et al.
4,941,170	A	7/1990	Herbst
4,942,598	A	7/1990	Davis
4,953,204	A	8/1990	Cuschleg, Jr. et al.
4,955,047	A	9/1990	Morganstein et al.
4,956,835	A	9/1990	Grover
4,959,854	A	9/1990	Cave et al.
4,967,288	A	10/1990	Mizutori et al.
4,969,184	A	11/1990	Gordon et al.
4,972,462	A	11/1990	Shibata
4,974,254	A	11/1990	Perine et al.
4,975,941	A	12/1990	Morganstein et al.
4,985,913	A	1/1991	Shalom et al.
4,994,926	A	2/1991	Gordon et al.
4,996,704	A	2/1991	Brunson
5,003,575	A	3/1991	Chamberlin et al.
5,003,577	A	3/1991	Ertz et al.
5,008,926	A	4/1991	Misholi
5,020,095	A	5/1991	Morganstein et al.
5,027,384	A	6/1991	Morganstein
5,029,196	A	7/1991	Morganstein
5,036,533	A	7/1991	Carter et al.
5,054,054	A	10/1991	Pessia et al.
5,065,254	A	11/1991	Hishida
5,086,385	A	2/1992	Launey et al.
5,095,445	A	3/1992	Sekiguchi
5,099,509	A	3/1992	Morganstein et al.
5,109,405	A	4/1992	Morganstein
5,128,984	A	7/1992	Katz
5,131,024	A	7/1992	Pugh et al.
5,133,004	A	7/1992	Heileman, Jr. et al.
5,145,452	A	9/1992	Chevalier
5,166,974	A	11/1992	Morganstein et al.
5,179,585	A	1/1993	MacMillan, Jr. et al.
5,193,110	A	3/1993	Jones et al.
5,195,086	A	3/1993	Baumgartner et al.
5,233,600	A	8/1993	Pekarske
5,243,643	A	9/1993	Sattar et al.
5,243,645	A	9/1993	Bissell et al.
5,249,219	A	9/1993	Morganstein et al.
5,263,084	A	11/1993	Chaput et al.
5,291,302	A	3/1994	Gordon et al.
5,291,479	A	3/1994	Vaziri et al.
5,303,298	A	4/1994	Morganstein et al.
5,307,399	A	4/1994	Dai et al.
5,309,504	A	5/1994	Morganstein
5,325,421	A	6/1994	Hou et al.
5,327,486	A	7/1994	Wolff et al.
5,327,529	A	7/1994	Fults et al.
5,329,578	A	7/1994	Brennan et al.
5,333,266	A	7/1994	Boaz et al.
5,347,574	A	9/1994	Morganstein
5,355,403	A	10/1994	Richardson, Jr. et al.
5,359,598	A	10/1994	Steagall et al.
5,365,524	A	11/1994	Hiller et al.
5,375,161	A	12/1994	Fuller et al.
5,384,771	A	1/1995	Isidoro et al.
5,404,231	A	4/1995	Bloomfield
5,408,526	A	4/1995	McFarland et al.
5,414,754	A	5/1995	Pugh et al.
5,416,834	A	5/1995	Bales et al.
5,432,845	A	7/1995	Burd et al.
5,436,963	A	7/1995	Fitzpatrick et al.
5,459,584	A	10/1995	Gordon et al.
5,463,684	A	10/1995	Morduch et al.
5,475,791	A	12/1995	Schalk et al.
5,479,487	A	12/1995	Hammond
5,495,484	A	2/1996	Self et al.
5,497,373	A	3/1996	Hulen et al.
5,499,288	A	3/1996	Hunt et al.
5,515,427	A	5/1996	Carlsen et al.
5,517,558	A	5/1996	Schalk
5,526,353	A	6/1996	Henley et al.
5,533,115	A	7/1996	Hollenbach et al.
5,537,461	A	7/1996	Bridges et al.
5,555,100	A	9/1996	Bloomfield et al.
5,559,611	A	9/1996	Bloomfield et al.
5,559,859	A	9/1996	Dai et al.
5,566,236	A	10/1996	MeLampy et al.
5,603,031	A	2/1997	White et al.
5,608,786	A	3/1997	Gordon
5,610,910	A	3/1997	Focsaneanu et al.
5,610,970	A	3/1997	Fuller et al.
5,611,031	A	3/1997	Hertzfield et al.
5,630,079	A	5/1997	McLaughlin
5,652,789	A	7/1997	Miner et al.
5,657,376	A	8/1997	Espeut et al.
5,659,597	A	8/1997	Bareis et al.
5,666,401	A	9/1997	Morganstein et al.
5,675,507	A	10/1997	Bobo, II
5,675,811	A	10/1997	Broedner et al.
5,689,669	A	11/1997	Lynch et al.
5,692,187	A	11/1997	Goldman et al.
5,699,486	A	12/1997	Tullis et al.
5,712,903	A	1/1998	Bartholomew et al.
5,719,921	A	2/1998	Vysotsky et al.
5,721,908	A	2/1998	Lagarde et al.
5,724,408	A	3/1998	Morganstein
5,742,905	A	4/1998	Pepe et al.
5,752,191	A	5/1998	Fuller et al.
5,764,639	A	6/1998	Staples et al.
5,764,736	A	6/1998	Shachar et al.
5,764,910	A	6/1998	Shachar
5,787,298	A	7/1998	Broedner et al.
5,793,993	A	8/1998	Broedner et al.
5,799,065	A	8/1998	Junqua et al.
5,809,282	A	9/1998	Cooper et al.
5,812,796	A	9/1998	Broedner et al.
5,819,220	A	10/1998	Sarukkai et al.
5,819,306	A	10/1998	Goldman et al.
5,822,727	A	10/1998	Garberg et al.

5,832,063	A	11/1998	Vysotsky et al.
5,835,570	A	11/1998	Wattenbarger
5,838,682	A	11/1998	Dekelbaum et al.
5,867,494	A	2/1999	Krishnaswamy et al.
5,867,495	A	2/1999	Elliott et al.
5,873,080	A	2/1999	Coden et al.
5,881,134	A	3/1999	Foster et al.
5,884,032	A	3/1999	Bateman et al.
5,884,262	A	3/1999	Wise et al.
5,884,266	A	3/1999	Dvorak
5,890,123	A	3/1999	Brown et al.
5,915,001	A	6/1999	Uppaluru
5,917,817	A	6/1999	Dunn et al.
5,943,399	A	8/1999	Bannister et al.
5,946,389	A	8/1999	Dold
5,953,392	A	9/1999	Rhie et al.
5,974,413	A	10/1999	Beauregard et al.
5,999,525	A	12/1999	Krishnaswamy et al.
6,012,088	A	1/2000	Li et al.
6,014,437	A	1/2000	Acker et al.
6,018,710	A	1/2000	Wynblatt et al.
6,021,181	A	2/2000	Miner et al.
6,031,904	A	2/2000	An et al.
6,038,305	A	3/2000	McAllister et al.
6,047,053	A	4/2000	Miner et al.
6,067,516	A	5/2000	Levay et al.
6,078,580	A	6/2000	Mandalia et al.
6,081,518	A	6/2000	Bowman-Amuah
6,081,782	A	6/2000	Rabin
6,091,808	A	7/2000	Wood et al.
6,101,472	A	8/2000	Giangarra et al.
6,104,803	A	8/2000	Weser et al.
6,115,742	A	9/2000	Franklin et al.
6,130,933	A	10/2000	Miloslavsky
6,157,705	A	12/2000	Perrone
6,185,535	B1	2/2001	Hedin et al.
6,195,357	B1	2/2001	Polcyn
6,208,638	B1	3/2001	Rieley et al.
6,230,132	B1	5/2001	Class et al.
6,233,318	B1	5/2001	Picard et al.
6,243,373	B1	6/2001	Turock
6,252,944	B1	6/2001	Hansen, II et al.
6,269,336	B1	7/2001	Ladd et al.
6,285,745	B1	9/2001	Bartholomew et al.
6,327,572	B1	12/2001	Morton et al.
6,349,132	B1	2/2002	Wesemann et al.
6,353,661	B1	3/2002	Bailey, III
6,366,578	B1	4/2002	Johnson
6,424,945	B1	7/2002	Sorsa
6,434,529	B1	8/2002	Walker et al.
6,446,076	B1	9/2002	Burkey et al.
6,456,699	B1	9/2002	Burg et al.
6,477,420	B1	11/2002	Struble et al.
6,501,966	B1	12/2002	Bareis et al.
6,505,163	B1	1/2003	Zhang et al.
6,529,948	B1	3/2003	Bowman-Amuah
6,532,444	B1	3/2003	Weber
6,539,359	B1	3/2003	Ladd et al.
6,546,393	B1	4/2003	Khan
6,584,439	B1	6/2003	Geilhufe et al.
6,594,348	B1	7/2003	Bjurstrom et al.
6,618,726	B1	9/2003	Colbath et al.
6,636,831	B1	10/2003	Profit, Jr. et al.
6,665,640	B1	12/2003	Bennett et al.
6,687,341	B1	2/2004	Koch et al.
6,718,015	B1	4/2004	Berstis
6,721,705	B2	4/2004	Kurganov et al.
6,732,142	B1	5/2004	Bates et al.
6,771,732	B2	8/2004	Xiao et al.
6,775,264	B1	8/2004	Kurganov
6,807,257	B1	10/2004	Kurganov
6,823,370	B1	11/2004	Kredo et al.
6,888,929	B1	5/2005	Saylor et al.
6,922,733	B1	7/2005	Kuiken et al.
6,941,273	B1	9/2005	Loghmani et al.
6,964,012	B1	11/2005	Zirngibl et al.
6,964,023	B2	11/2005	Maes et al.
6,965,864	B1	11/2005	Thrift et al.
6,996,609	B2	2/2006	Hickman et al.
7,003,463	B1	2/2006	Maes et al.
7,050,977	B1	5/2006	Bennett
7,076,431	B2	7/2006	Kurganov et al.
7,386,455	B2	6/2008	Kurganov et al.
7,516,190	B2	4/2009	Kurganov
2001/0032234	A1	10/2001	Summers et al.
2001/0040885	A1	11/2001	Jonas et al.
2001/0048676	A1	12/2001	Jimenez et al.
2002/0006126	A1	1/2002	Johnson et al.

FOREIGN PATENT DOCUMENTS

EP	0 572 544	B1	9/1996
GB	2 211 698	A	7/1989
GB	2 240 693	A	8/1991
GB	2 317 782	A	1/1998
JP	1-258526		10/1989
WO	WO 91/07838		5/1991
WO	WO 91/18466		11/1991
WO	WO 96/09710		3/1996
WO	WO 97/37481		10/1997
WO	WO 98/23058		5/1998

OTHER PUBLICATIONS

Amended Complaint, *Parus Holdings. Inc*. v. *Web Telephony LLC & Robert Swartz*, Case No. 06-cv-01146 (N.D. III.),, Jul. 10, 2006, 14 pages.

AT&T, Press Release, "AT&T Customers Can Teach Systems to Listen and Respond to Voice", Jan. 17, 1995, pp. 1-2, Basking Ridge, NJ., available at www.lucent.com/press/0195/950117.gbb.html (accessed Mar. 15, 2005).

Bellcore Technology Licensing, "The Electronic Receptionist—A Knowledge-Based Approach to Personal Communications", 1994, pp. 1-8.

Brachman et al., "Fragmentation in Store-and-Forward Message Transfer", IEEE Communications Magazine, vol. 26(7), Jul. 1998, pp. 18-27.

"Business Phone Systems for Advanced Offices", NTT Review, vol. 2 (6), Nov. 1990, pp. 52-54.

Cole et al., "An Architecture for a Mobile OSI Mail Access System", IEEE Journal on Selected Areas in Communications, vol. 7 (2), Feb. 1989, pp. 249-256.

"Data Communications Networks: Message Handling Systems", Fasciele, VIII. 7-Recommendations X.400-X.430, 38 pages, date unknown.

DAX Systems, Inc., Press Release, "Speech Recognition Success in DAX's Grasp", Nov. 22, 1995, pp. 1-2, Pine Brook, NJ.

Defendants Answer to the Amended Complaint and Demand for Jury Trial, *Parus Holdings, Inc*. v. *Web Telephone LLC & Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Aug. 10, 2006, 14 pages.

Faxpak Store and Forward Facsimile Transmission Service, Electrical Communication, vol. 54 (3), 1979, pp. 251-255.

Garcia et al., "Issues in Multimedia Computer-Based Message Systems Design and Standardization", NATO ASI Series, vol. 1-6, 1984, 18 pgs.

"Globecom '85 IEEE Global Telecommunications Conference, " New Orleans, LA., Dec. 2-5, 1985, pp. 1295-1300.

Hemphill et al., "Speech-Aware Multimedia," *IEEE MultiMedia*, Spring 1996, vol. 3, No. 1, pp. 74-78, IEEE. As indicated on the cover page of the journal, which is attached hereto as Attachment 4, the reference was received by Cornell University on Mar. 25, 1996.

Hunt et al., "Long-Distance Remote Control to the Rescue", Chicago Tribune, Jun. 15, 2002, Section 4, p. 15.

"Introducing PIC SuperFax, First PC/Fax System to Run Under Windows", Pacific Image Communications, Pasadena, CA, Date Unknown, 4 pgs.

"Juggler by PureSpeech", p. 1, available at http://members.aol.com/compqanda1/juggler.html (accessed on Dec. 8, 2006).

Kubala et al., "BYBLOS Speech Recognition Benchmark Results", *Workshop on Speech & Natural Language*, Feb. 19-22, 1991. According to the web site http://portal.acm.org/citation.cfm?id=112405.112415&coll..., attached hereto as Attachment 3, the reference was published in 1991, Morgan Kaufman Publishers, San Franscisco, CA. The distribution date is not presently known.

Ly, "Chatter: A Conversational Telephone Agent", submitted to Program in Media Arts & Sciences, MIT, 1993, pp. 1-130.
Maeda, et al., "An Intelligent Customer-Controlled Switching System", IEEE Global Telecommunications Conference, Hollywood, Florida, Nov. 28-Dec. 1, 1988, pp. 1499-1503.
Markowitz, J., "The Ultimate Computer Input Device May Be Right Under Your Nose", *Byte*, Dec. 1995, pp. 1-13, available at www.byte.com/art/9512/sec8/art1.htm (accessed Mar. 15, 2005).
Marx et al., "Mail Call: Message Presentation and Navigation in a Nonvisual Environment," *SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, B.C., Canada, Apr. 13-18, 1996. As shown on Attachment 2, the web site http://www.usabilityviews.com/uv001673.html shows a date of Apr. 16, 1996. The distribution date is not presently known.
Marx, M., "Toward Effective Conversational Messaging" (Thesis). As indicated on the cover page, the thesis was presented to the Departmental Committee on Graduate Students, Program in Media Arts and Sciences, School of Architecture and Planning, Massachusetts Institute of Technology on May 12, 1995. According to the web site http://www.thesis.mit.edu/Dienst/Repository/2.0/Body/0018.mit.theses/1995-314/rfc1807bib, attached hereto as Attachment 1, the thesis was indexed on Mar. 21, 2000.
Oye, Phil, "Juggler", p. 1, available at http://www.philove.com/work/juggler/index.shtml (accessed on Dec. 8, 2006).
Oye, Phil, "Juggler", p. 1, available at http://www.philoye.com/work/juggler_2.shtml (accessed on Dec. 8, 2006).
Oye, Phil, "Juggler", p. 1, available at http://www.philoye.com/work/juggler_3.shtml (accessed on Dec. 8, 2006).
Perdue et al., "Conversant® 1 Voice System: Architecture and Applications", Jul. 17, 1986, AT&T Technical Journal, pp. 1-14.
Plaintiff Parus Holdings, Inc.'s Supplemental Responses to Defendant Web Telephone LLC's First Set of Interrogatories (Nos. 1-12), *Parus Holdings, Inc*. v. *Web Telephony LLC Y Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Oct. 31, 2006, 32 pages.
Plaintiff Parus Holdings, Inc.'s Supplemental Responses to Defendant Web Telephony LLC's Second Set of Interrogatories (Nos. 13-17), *Parus Holdings,Inc*. v. *Web Telephony LLC & Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Oct. 31, 2006, 31 pages.
Print outs of Internet web site, "Wildfire Communications, Inc.,", Nov. 5, 1997, including print outs of the following web pages: http://www.wildfire.com (1 pg); http://www.wildfire.com/consumerhome.html (2 pgs.); http://www.wildfire.com/106.html (2 pgs.); http://www.wildfire.com/carrierhome.html (2 pgs.); http://www.wildfire.com/sfandb.html (3 pgs.); http://www.wildfire.com/about.html (1 pg.); http://www.wildfire.com/abtmgmt.html (3 pgs); http://www.wildfire.com/scoop.html (2 pgs.); and http://www.wildfire.com/intel.html (1 pg.); and http://www.wildfire.com/msft.html (2 pgs).
"Proceedings of the IFIP World Computer Congress", Dublin, Ireland, Sep. 1-5, 1986.
"PureSpeech Announces Juggler PC Systemfor First Quarter of 1997", *HighBeam Research*, Sep. 19, 1996, pp. 1-3, available at http://www.highbeam.com/doc/1G1-186909545.html (accessed on Dec. 8, 2006).
"PureSpeech's Juggler", *Teleconnect*, Dec. 1996 issue, p. 36.
PureSpeech, "Meet the Voice of Juggler!", pp. 1-3, the date of Nov. 18, 1996 is shown at the top of p. 1.
Ross, randy, "Retrieve E-mail from a Telephone", Oct. 7, 1996, pp. 1-2, available at http://resna.org/ProfessOrg?Sigs?SIGSites/sig11/archive/juggler.htm (accessed on Dec. 8, 2006). Printout indicates that the article was originally printed in *PC World*.
Sartori, M., "Speech Recognition", Apr. 1995, pp. 1-9, Mercury Communications, available at www.gar.co.uk/technology_watch/speech.htm (accessed Mar. 15, 2005).
Schmandt et al., "A Conversational Telephone Messaging Systems", IEEE Transactions on Consumer Electronics, 1984, vol. CE-30, No. 3, pp. xxi-xxiv.
Schmandt et al., "Phone Shell: The Telephone as Computer Terminal", ACM Multimedia, 1993, 11 pgs.
Schmandt et al., "Phone Slave: A Graphical Telecommunictions Interface", Proceedings of the SID, 1985, vol. 26/1, pp. 79-82.
"Secretarial Branch Exchanged", IBM Technical Disclosure Bulletin, vol. 26 (5), Oct. 1983, pp. 2645-2647.
Shimamura, et al., "Review of the Electrical Communication Laboratories", vol. 418 (33), No. 1, Tokyo, Japan, 1985, pp. 31-39.
"The VMX Systems Product Reference Manual: Product Description Volume", May 1994, vol. 1, release 7.1, VMX, Inc. (Octel Communications Corp.) San jose, CA USA.
"VMXworks Product Reference Manual: vol. 3 Programmer's Guide", Jul. 1994, vols. 3 & 4, Release 3.1, Octel Communications corp., Milpitas, CA, USA.
"Wildfire Communication, Inc.", Harvard Business School, Mar. 21, 1996, Publ. No. 9-396-305, pp. 1-22.
"WordPerfect: New Telephony Features Boost Office", WordPerfect Office TechBrief, 1994, Info-World Publishing. Co., vol. 10, Issue 2, pp. 2-3.
Yang, C., "INETPhone—Telephone Services and Servers on the Internet", Apr. 1995, University of North Texas, pp. 1-6.

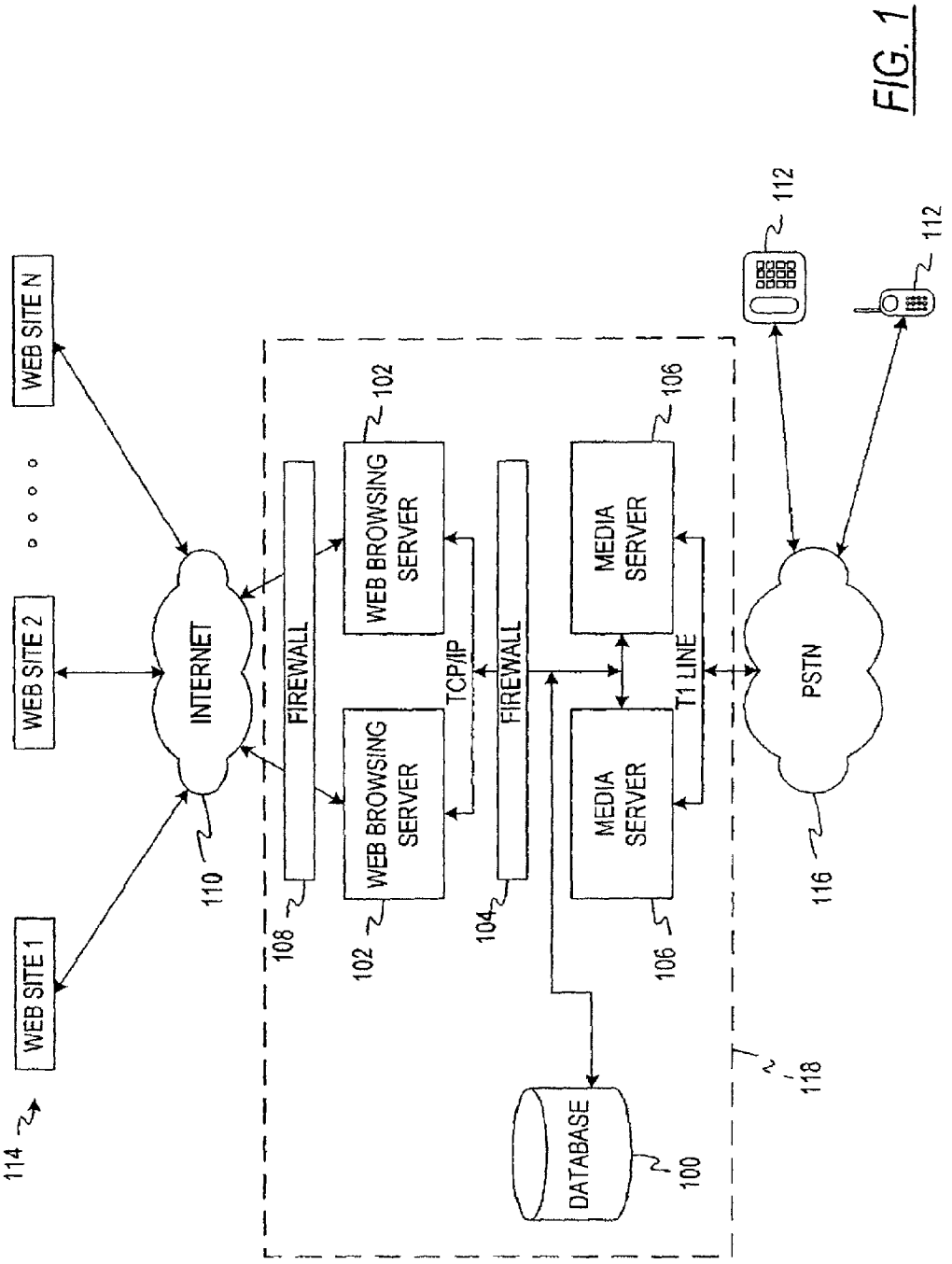

WEB SITE N
WEB SITE 2
WEB SITE 1
INTERNET
FIREWALL
WEB BROWSING SERVER
WEB BROWSING SERVER
TCP/IP
FIREWALL
MEDIA SERVER
MEDIA SERVER
T1 LINE
PSTN
DATABASE
110
108
102
102
104
106
106
112
112
116
118
114
100

FIG. 1

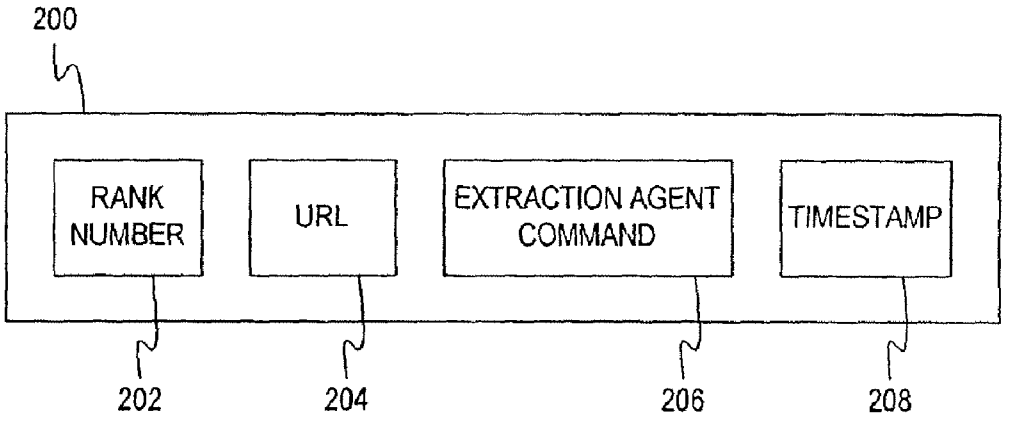

200
RANK NUMBER
URL
EXTRACTION AGENT COMMAND
TIMESTAMP
202
204
206
208

FIG. 2

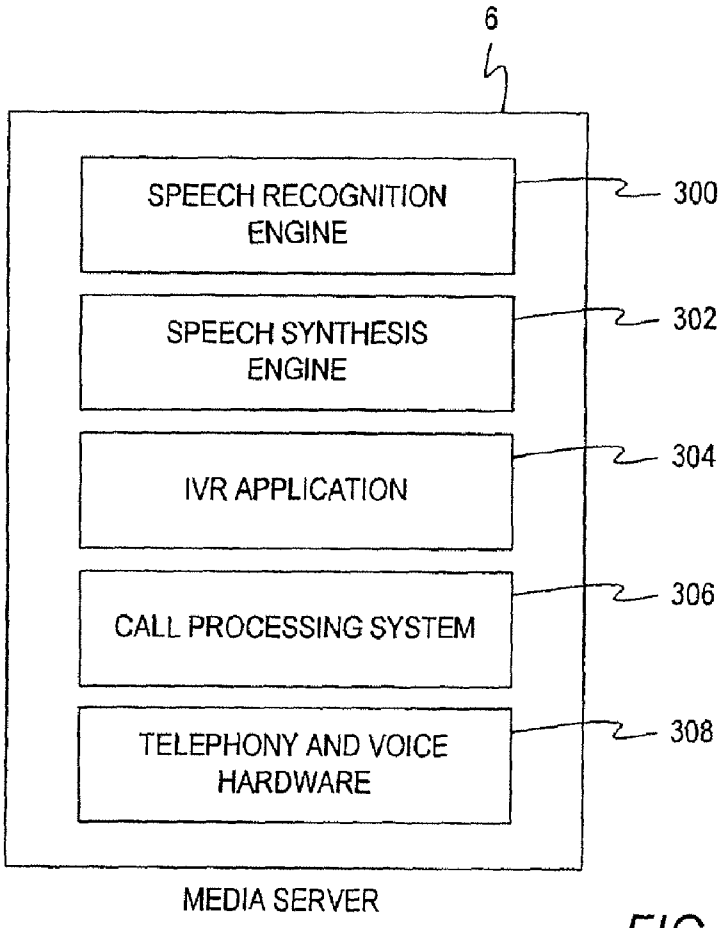

6
SPEECH RECOGNITION ENGINE
300
SPEECH SYNTHESIS ENGINE
302
IVR APPLICATION
304
CALL PROCESSING SYSTEM
306
TELEPHONY AND VOICE HARDWARE
308
MEDIA SERVER

FIG. 3

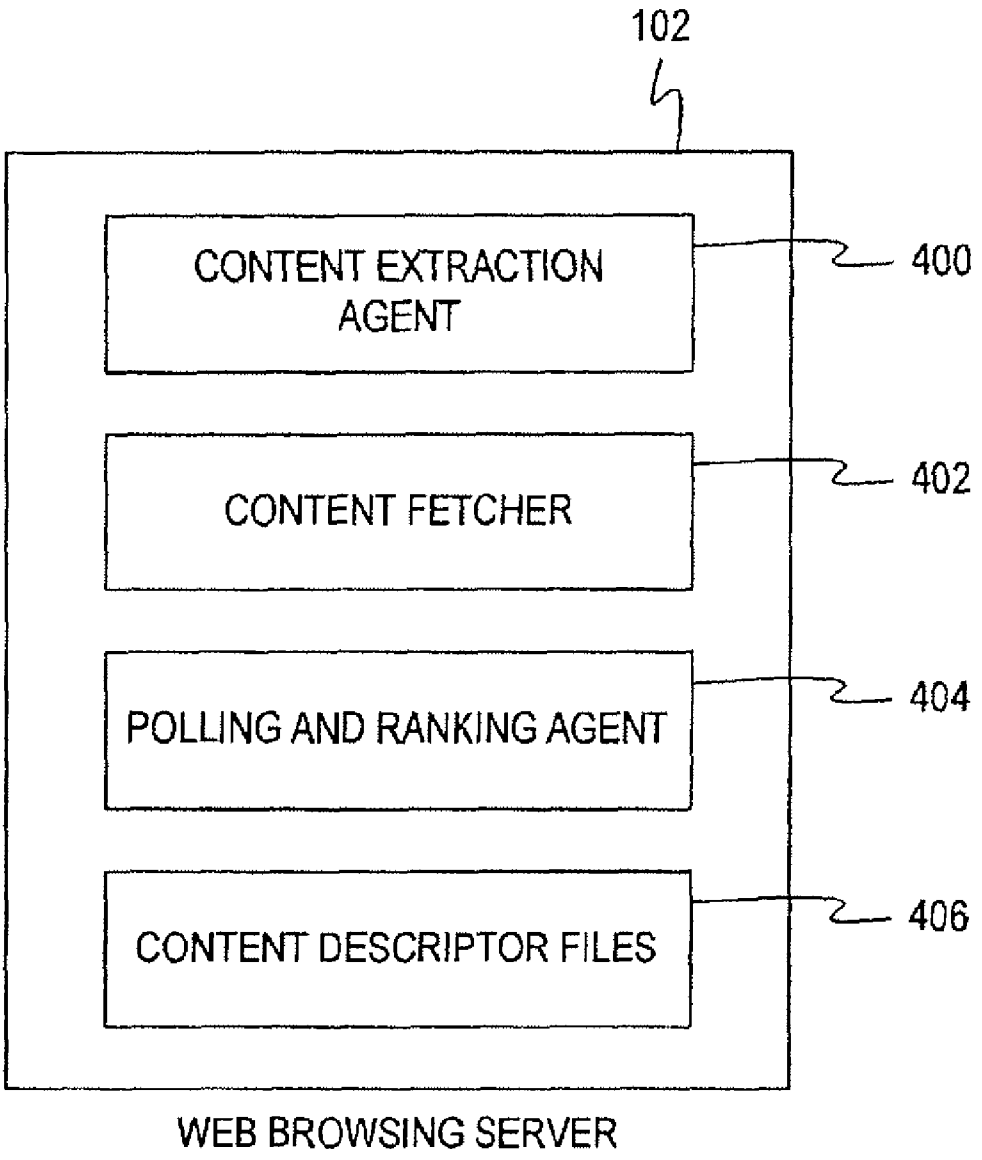

102
CONTENT EXTRACTION AGENT
400
CONTENT FETCHER
402
POLLING AND RANKING AGENT
404
CONTENT DESCRIPTOR FILES
406
WEB BROWSING SERVER

FIG. 4

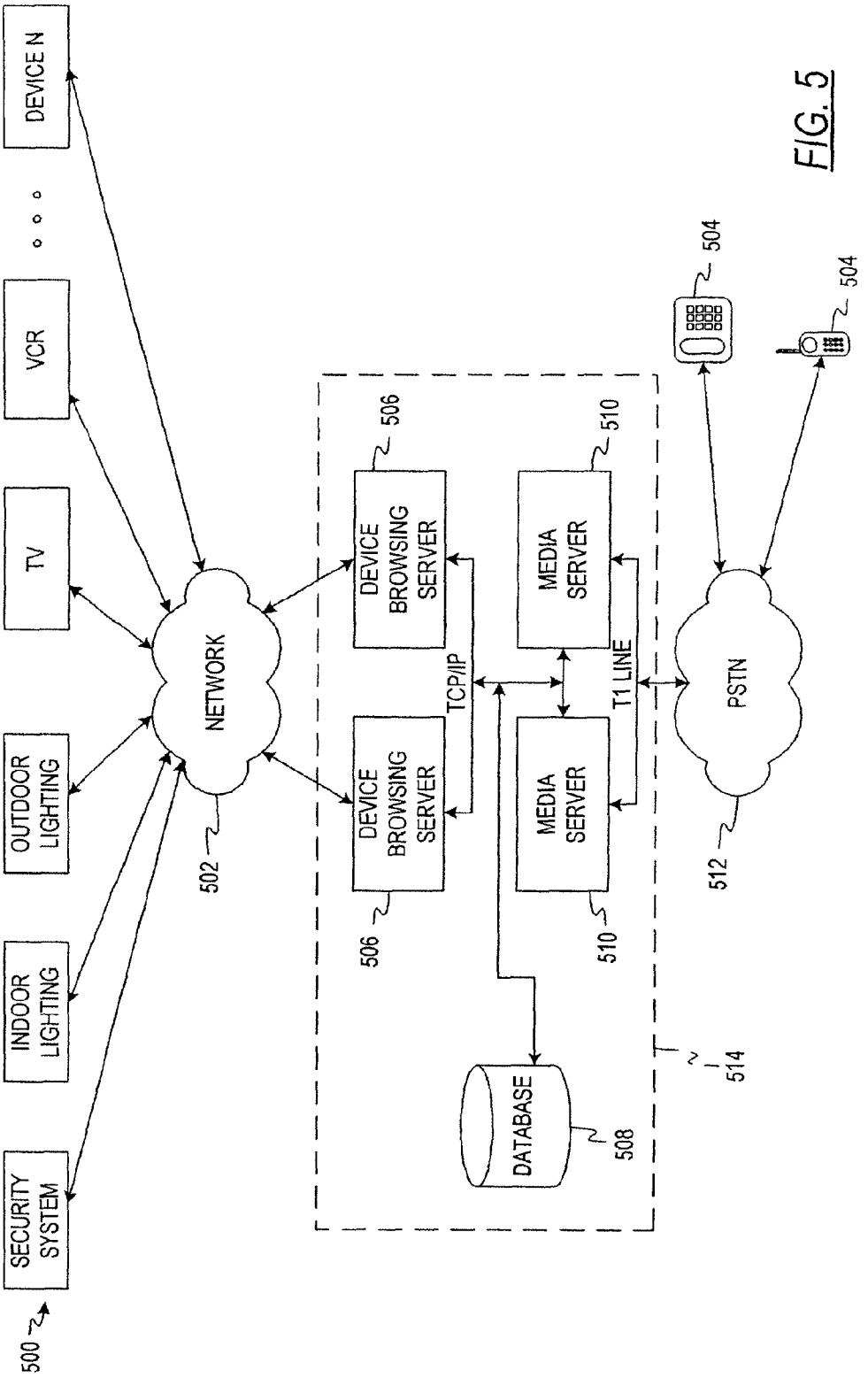

DEVICE N
VCR
TV
OUTDOOR LIGHTING
INDOOR LIGHTING
SECURITY SYSTEM
500
NETWORK
502
DEVICE BROWSING SERVER
506
DEVICE BROWSING SERVER
506
TCP/IP
MEDIA SERVER
510
MEDIA SERVER
510
T1 LINE
DATABASE
508
514
PSTN
512
504
504

FIG. 5

ROBUST VOICE BROWSER SYSTEM AND VOICE ACTIVATED DEVICE CONTROLLER

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 12/030,556, filed Feb. 13, 2008, now allowed, which is a continuation of U.S. patent application Ser. No. 11/409,703, filed Apr. 24, 2006 and issued as U.S. Pat. No. 7,386,455 on Jun. 10, 2008, which is a continuation of U.S. patent application Ser. No. 10/821,690, filed Apr. 9, 2004 and issued as U.S. Pat. No. 7,076,431 on Jul. 11, 2006, which is a continuation of U.S. patent application Ser. No. 09/776,996, filed Feb. 5, 2001 and issued as U.S. Pat. No. 6,721,705 on Apr. 13, 2004, which claims the benefit of priority to U.S. Provisional Application No. 60/180,344, filed Feb. 4, 2000, entitled "Voice Activated Information Retrieval System" and U.S. Provisional Patent Application No. 60/233,068, filed Sep. 15, 2000, entitled "Robust Voice Browser System and Voice Activated Device Controller", all of which are herein incorporated by reference in their entirety.

FIELD OF THE INVENTION

The present invention relates to a robust and highly reliable system that allows users to browse web sites and retrieve information by using conversational voice commands. Additionally, the present invention allows users to control and monitor other systems and devices that are connected the Internet or any other network by using voice commands.

BACKGROUND OF THE INVENTION

Currently, three options exist for a user who wishes to gather information from a web site accessible over the Internet. The first option is to use a desktop or a laptop computer connected to a telephone line via a modem or connected to a network with Internet access. The second option is to use a Personal Digital Assistant (PDA) that has the capability of connecting to the Internet either through a modem or a wireless connection. The third option is to use one of the newly designed web-phones or web-pagers that are now being offered on the market. Although each of these options provide a user with access to the Internet to browse web sites, each of them have their own drawbacks.

Desktop computers are very large and bulky and are difficult to transport. Laptop computers solve this inconvenience, but many are still quite heavy and are inconvenient to carry. Further, laptop computers cannot be carried and used everywhere a user travels. For instance, if a user wishes to obtain information from a remote location where no electricity or communication lines are installed, it would be nearly impossible to use a laptop computer. Oftentimes, information is needed on an immediate basis where a computer is not accessible. Furthermore, the use of laptop or desktop computers to access the Internet requires either a direct or a dial-up connection tan an Internet Service Provider (ISP). Oftentimes, such connections are not available when a user desires to connect to the Internet to acquire information.

The second option for remotely accessing web sites is the use of PDAs. These devices also have their own set of drawbacks. First, PDAs with the ability to connect to the Internet and access web sites are not readily available. As a result, these PDAs tend to be very expensive. Furthermore, users are usually required to pay a special service fee to enable the web browsing feature of the PDA. A further disadvantage of these

PDAS is that web sites must be specifically designed to allow these devices to access information on the web site. Therefore, a limited number of web sites are available that are accessible by these web-enabled PDAs. Finally, it is very common today for users to carry cell phones, however, users must also carry a separate PDA if they require the ability to gather information from various web sites. Users are therefore subjected to added expenses since they must pay for both cellular telephone service and also for the web-enabling service for the PDA. This results in a very expensive alternative for the consumer.

The third alternative mentioned above is the use of web-phones or web-pagers. These devices suffer many of the same drawbacks as PDAs. First, these devices are expensive to purchase. Further, the number of web sites accessible to these devices is limited since web sites must be specifically designed to allow access by these devices. Furthermore, users are often required to pay an additional fee in order to gain wireless web access. Again, this service is expensive. Another drawback of these web-phones or web-pagers is that as technology develops, the methods used by the various web sites to allow access by these devices may change. These changes may require users to purchase new web-phones or web-pagers or have the current device serviced in order to upgrade the firmware or operating system stored within the device. At the least, this would be inconvenient to users and may actually be quite expensive.

Therefore, a need exists for a system that allows users to easily access and browse the Internet from any location. Such a system would only require users to have access to any type of telephone and would not require users to subscribe to multiple services.

In the rapidly changing area of Internet applications, web sites change frequently. The design of the web site may change, the information required by the web site in order to perform searches may change, and the method of reporting search results may change. Web browsing applications that submit search requests and interpret responses from these web sites based upon expected formats will produce errors and useless responses when such changes occur. Therefore, a need exists for a system that can detect modifications to web sites and adapt to such changes in order to quickly and accurately provide the information requested by a user through a voice enabled device, such as a telephone.

When users access web sites using devices such as personal computers, delays in receiving responses are tolerated and are even expected, however, such delays are not expected when a user communicates with a telephone. Users expect communications over a telephone to occur immediately with a minimal amount of delay time. A user attempting to find information using a telephone expects immediate responses to his search requests. A system that introduces too much delay between the time a user makes a request and the time of response will not be tolerated by users and will lose its usefulness. Therefore, it is important that a voice browsing system that uses telephonic communications selects web sites that provide rapid responses since speed is an important factor for maintaining the system's desirability and usability. Therefore, a need exists for a system that accesses web sites based upon their speed of operation.

SUMMARY OF THE INVENTION

It is an object of an embodiment of the present invention to allow users to gather information from web sites by using voice enabled devices, such as wireline or wireless telephones.

An additional object of an embodiment of the present invention is to provide a system and method that allows the searching and retrieving of publicly available information by controlling a web browsing server using naturally spoken voice commands.

It is an object of another embodiment of the present invention to provide a robust voice browsing system that can obtain the same information from several web sites based upon a ranking order. The ranking order is automatically adjusted if the system detects that a given web site is not functioning, is too slow, or has been modified in such a way that the requested information cannot be retrieved any longer.

A still further object of an embodiment of the present invention is to allow users to gather information from web sites from any location where a telephonic connection can be made.

Another object of an embodiment of the present invention is to allows users to browse web sites on the Internet using conversational voice commands spoken into wireless or wireline telephones or other voice enabled devices.

An additional object an embodiment of the present invention is to provide a system and method for using voice commands to control and monitor devices connected to a network.

It is an object of an embodiment of the present invention to provide a system and method which allows devices connected to a network to be controlled by conversational voice commands spoken into any voice enabled device interconnected with the same network.

The present invention relates to a system for acquiring information from sources on a network, such as the Internet. A voice browsing system maintains a database containing a list of information sources, such as web sites, connected to a network. Each of the information sources is assigned a rank number which is listed in the database along with the record for the information source. In response to a speech command received from a user, a network interface system accesses the information source with the highest rank number in order to retrieve information requested by the user.

The a preferred embodiment of the present invention allows users to access and browse web sites when they do not have access to computers with Internet access. This is accomplished by providing a voice browsing system and method that allows users to browse web sites using conversational voice commands spoken into any type of voice enabled device (i.e., any type of wireline or wireless telephone, IP phone, wireless PDA, or other wireless device). These spoken commands are then converted into data messages by a speech recognition software engine running on a user interface system. These data messages are then sent to and processed by a network interface system. This network interface system then generates the proper requests that are transmitted to the desired web site over the Internet. Responses sent from the web site are received and processed by the network interface system and then converted into an audio message via a speech synthesis engine or a pre-recorded audio concatenation application and finally transmitted to the user's voice enabled device.

A preferred embodiment of the voice browser system and method uses a web site polling and ranking methodology that allows the system to detect changes in web sites and adapt to those changes in real-time. This enables the voice browser system of a preferred embodiment to deliver highly reliable information to users over any voice enabled device. This ranking system also enables the present invention to provide rapid responses to user requests. Long delays before receiving responses to requests are not tolerated by users of voice-based systems, such as telephones. When a user speaks into a telephone, an almost immediate response is expected. This expectation does not exist for non-voice communications, such as email transmissions or accessing a web site using a personal computer. In such situations, a reasonable amount of transmission delay is acceptable. The ranking system of implemented by a preferred embodiment of the present invention ensures users will always receive the fastest possible response to their request.

An alternative embodiment of the present invention allows users to control and monitor the operation of a variety of household devices connected to a network using speech commands spoken into a voice enabled device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a depiction of the voice browsing system of the first embodiment of the present invention;

FIG. **2** is a block diagram of a database record used by the first preferred embodiment of the present invention;

FIG. **3** is a block diagram of a media server used by the preferred embodiment;

FIG. **4** is a block diagram of a web browsing server used by the preferred embodiment; and

FIG. **5** is a depiction of the device browsing system of the second embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A first embodiment of the present invention is a system and method for allowing users to browse information sources, such as web sites, by using naturally spoken, conversational voice commands spoken into a voice enabled device. Users are not required to learn a special language or command set in order to communicate with the voice browsing system of the present invention. Common and ordinary commands and phrases are all that is required for a user to operate the voice browsing system. The voice browsing system recognizes naturally spoken voice commands and is speaker-independent; it does not have to be trained to recognize the voice patterns of each individual user. Such speech recognition systems use phonemes to recognize spoken words and not predefined voice patterns.

The first embodiment allows users to select from various categories of information and to search those categories for desired data by using conversational voice commands. The voice browsing system of the first preferred embodiment includes a user interface system referred to as a media server. The media server contains a speech recognition software engine. This speech recognition engine is used to recognize natural, conversational voice commands spoken by the user and converts them into data messages based on the available recognition grammar. These data messages are then sent to a network interface system. In the first preferred embodiment, the network interface system is referred to as a web browsing server. The web browsing server then accesses the appropriate information source, such as a web site, to gather information requested by the user.

Responses received from the information sources are then transferred to the media server where speech synthesis engine converts the responses into audio messages that are transmitted to the user. A more detailed description of this embodiment will now be provided.

Referring to FIG. **1**, a database **100** designed by Webley Systems Incorporated is connected to one or more web browsing servers **102** as well as to one or more media servers **106**. The database may store information on magnetic media,

such as a hard disk drive, or it may store information via other widely acceptable methods for storing data, such as optical disks. The database **100** contains a separate set of records for each web site accessible by the system. An example of a web site record is shown in FIG. **2**. Each web site record **200** contains the rank number of the web site **202**, the associated Uniform Resource Locator (URL) **204**, and a command that enables the appropriate "extraction agent" **206** that is required in order to generate proper requests sent to and to format data received from the web site. The database record **200** also contains the timestamp **208** indicating the last time the web site was accessed. The extraction agent is described in more detail below. The database **100** categorizes each database record **200** according to the type of information provided by each web site. For instance, a first category of database records **200** may correspond to web sites that provide "weather" information. The database **100** may also contain a second category of records **200** for web sites that provide "stock" information. These categories may be further divided into subcategories. For instance, the "weather" category may contain subcategories depending upon type of weather information available to a user, such as "current weather" or "extended forecast". Within the "extended forecast" subcategory, a list of web site records may be stored that provide weather information for multiple days. The use of subcategories may allow the web browsing feature to provide more accurate, relevant, and up-to-date information to the user by accessing the most relevant web site. The number of records contained in each category or subcategory is not limited. In the preferred embodiment, three web site records are provided for each category.

Table 1 below depicts two database records **200** that are used with the preferred embodiment. These records also contain a field indicating the "category" of the record, which is "weather" in each of these examples.

TABLE 1

```
category:     weather
URL: URL=http: //cgi .cnn.com/cgi-
bin/weather/redirect? zip= zip
rank: 1
command:   web_dispatch .pl weather_cnn
browsingserver:        wportal1
browsingServerBackup:     wportal2
dateTime:    Dec 21 2000 2:15PM
category:     weather
URL: URL=http: //weather.lycos.com/wcfiveday.asp?city=zip
rank:          2
command:   web_dispatch.pl weather_lycos
browsingServer:        wportal1
browsingserverBackup:     wportal2
dateTime:    Dec 21 2000 1:45PM
```

The database also contains a listing of pre-recorded audio files used to create concatenated phrases and sentences. Further, database **100** may contain customer profile information, system activity reports, and any other data or software servers necessary for the testing or administration of the voice browsing system.

The operation of the media servers **106** will now be discussed in relation to FIG. **3**. The media servers **106** function as user interface systems. In the preferred embodiment, the media servers **106** contain a speech recognition engine **300**, a speech synthesis engine **302**, an Interactive Voice Response (IVR) application **304**, a call processing system **306**, and telephony and voice hardware **308** required to communicate with the Public Switched Telephone Network (PSTN) **116**. In the preferred embodiment, each media server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The speech recognition function is performed by a speech recognition engine **300** that converts voice commands received from the user's voice enabled device **112** (i.e., any type of wireline or wireless telephone, Internet Protocol (IP) phones, or other special wireless units) into data messages. In the preferred embodiment, voice commands and audio messages are transmitted using the PSTN **116** and data is transmitted using the TCP/IP communications protocol. However, one skilled in the art would recognize that other transmission protocols may be used for either voice or data. Other possible transmission protocols would include SIP/VoIP (Session Initiation Protocol/Voice over IP), Asynchronous Transfer Mode (ATM) and Frame Relay. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www.nuance.com). The Nuance engine capacity is measured in recognition units based on CPU type as defined in the vendor specification. The natural speech recognition grammars (i.e., what a user can say that will be recognized by the speech recognition engine) were developed by Webley Systems.

Table 2 below provides a partial source code listing of the recognition grammars used by the speech recognition engine of the preferred embodiment for obtaining weather information.

TABLE 2

```
?WHAT_IS ?the weather ?[info information report conditions]
        ? ( (?like in )
        ]
    UScities:n
        {<paraml $n.zip> <param2 $n.city> <param3
        $n.state>}
        ( (area code) AREA_CODE:n ) {<paraml $n>}
        ( AREA_CODE:n (area code) ) {<paraml $n>}
        ( (zip ?code) ZIP_CODE:n ) {<paraml $n>}
        ( ZIP_CODE:n (zip ?code) ) {<paraml $n>}
        ]
        )
            ) {<menu 194>}
```

The media server **106** uses recognition results generated by the speech recognition engine **300** to retrieve a web site record **200** stored in the database **100** that can provide the information requested by the user. The media server **106** processes the recognition result data identifying keywords that are used to search the web site records **200** contained in the database **100**. For instance, if the user's request was "What is the weather in Chicago?", the keywords "weather" and "Chicago" would be recognized. A web site record **200** with the highest rank number from the "weather" category within the database **100** would then be selected and transmitted to the web browsing server **102** along with an identifier indicating that Chicago weather is being requested.

The media servers **106** also contain a speech synthesis engine **302** that converts the data retrieved by the web browsing servers **102** into audio messages that are transmitted to the user's voice enabled device **112**. A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhsl.com).

A further description of the web browsing server **102** will be provided in relation to FIG. **4**. The web browsing servers **102** provide access to any computer network such as the Internet **110**. These servers are also capable of accessing databases stored on Local Area Networks (LANs) or Wide Area Networks (WANs). The web browsing servers receive

responses from web sites and extract the data requested by the user. This task is also known as "content extraction." The web browsing servers **102** also perform the task of periodically polling or "pinging" various web sites and modifying the ranking numbers of these web sites depending upon their response and speed. This polling feature is further discussed below. The web browsing server **102** is comprised of a content extraction agent **400**, a content fetcher **402**, a polling and ranking agent **404**, and the content descriptor files **406**. Each of these are software applications and will be discussed below.

Upon receiving a web site record **200** from the database **100** in response to a user request, the web browsing server **102** invokes the "content extraction agent" command **206** contained in the record **200**. The content extraction agent **400** allows the web browsing server **102** to properly format requests and read responses provided by the web site **114** identified in the URL field **204** of the web site record **200**. Each content extraction agent command **206** invokes the content extraction agent and identifies a content description file associated with the web page identified by the URL **204**. This content description the directs the extraction agent where to extract data from the accessed web page and how to format a response to the user utilizing that data. For example, the content description for a web page providing weather information would indicate where to insert the "city" name or ZIP code in order to retrieve Chicago weather information. Additionally, the content description file for each supported URL indicates the location on the web page where the response information is provided. The extraction agent **400** uses this information to properly extract from the web page the information requested by the user.

Table 3 below contains source code for a content extraction agent **400** used by the preferred embodiment.

TABLE 3

```
#! /usr/local/www/bin/sybper15
#$Header:
/usr/local/cvsroot/webley/agents/service/web_dispatch.pl, v
1.6
# Dispatches all web requests
#http://wcorp.itn.net/cgi/flstat?carrier=ua&flightno=155&mo
nabbr=jul&date=
6&stamp=OhLN~PdbuuE*itn/ord, itn/cb/sprint_hd
#http://cgi.cnnfn.com/flightview/rlm?airline=amt&number=300
require "config_tmp.pl";
# check parameters
die "Usage: $0 service [params]\n" if $#ARGV < 1;
#print STDERR @ARGV;
* get parameters
my ( $service, @param ) = @ARGV;
# check service
my %Services = (
                    weather_cnn => 'webget.pl weather_cnn',
                    weather lycos => 'webget.pl
weather lycos',
                    weather_weather => 'webget.pl
weather_weather',
                    weather_snap => 'webget.pl
weather_snap',
                    weather_infospace => 'webget.pl
weather_infospace',
                    stockQuote_yahoo => 'webget.pl stock',
                    flightstatusitn => 'webget.pl
flight_delay',
                    yellowPages_yahoo => 'yp_data.pl',
                    yellowPages_yahoo => 'yp_data.pl',
                    newsHeaders_newsreal => 'news.pl',
                    newsArticle_newsreal => 'news.pl',
# test param
my $date = 'date';
chop ( $date );
```

TABLE 3-continued

```
my ( $short_date ) = $date =~ /\s+(\w{3}\s+\d{1,2})\s+/;
my %Test = (
                    weather_cnn => '60053',
                    weather lycos => '60053',
                    weather_weather => '60053',
                    weather_snap => '60053',
                    weather_infospace => '60053',
                    stockQuote_yahoo => 'msft',
                    flightStatus_itn => 'ua 155 '.
$ short_date,
                    yellowPages_yahoo => 'tires 60015',
                    newsHeaders_newsreal => '1',
                    newsArticle_newsreal => '1 1',
die "$date: $0: error: no such service: $service (check this
script) \n"
unless $Services {$service };
# prepare absolute path to run other scripts
my ( $path, $script ) = $0 =~ m|(.*/) ([^/]*)|;
# store the service to compare against datatable
my $service_stored = $service;
# run service
while( !( $response = '$path$Services{$service }@param' ) )
{
        # response failed
        # check with test parameters
        $response = '$path$Services{$service }$Test{$service
}';
#       print "test: $path$Services{$service }$Test{$service
}";
        if ( $response ) {
                $service = &switch_service ( $service );
#               print "Wrong parameter values were supplied:
$service -
@param\n";
#               die "$date: $0: error: wrong parameters: $service
-
@param\n";
        }
        else {
                # change priority and notify
                $service = &increase_attempt( $service );
        }
}
# output the response
print $response;
sub increase_attempt {
        my ( $service ) = @_;
        my ( $service_name ) = split ( /_/, $service );
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                        . "set priority = ( select max( priority )
from
mcServiceRoute
                        . "where service = '$service_name' ) + 1,
"
                        . "date = getdate( ), "
                        . "attempt = attempt + 1 "
                        . "where route = '$script $service'" );
#       print "---$route===\n";
        # find new route
        my $route = @{&db_query( "select route from
mcServiceRoute "
                                . "where service =
'$service name'
                                . "and attempt < 5
"
                                . "order by
priority ")
                                }-> [ 0 ]{route };
        &db_query( "update mcServiceRoute "
                        . "set attempt = 0 "
                        . "where route = '$script $service'"
                if ( $route eq "$script $service"
                        or $route eq "$script $service_stored" );
        ( $service_name, $service ) = split( /\s+/, $route );
        die "$date: $0: error: no route for the service:
$service (add
```

TABLE 3-continued

```
more) \n"
                unless $service;
        return $service;
}
sub switch_service {
        my ( $service ) = @_;
        my ( $service_name ) = split( /_/, $service );
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                        . "set priority = ( select max ( priority )
from
mcServiceRoute "
                        . "where service = '$service_name' ) + 1,
"
                        . "date = getdate( ) "
                        "where route = '$script $service'" );
#       print "---$route===\n";
        # find new route
        my $route = @{&db_query( "select route from
mcServiceRoute "
                        . "where service =
'$service name' "
                        . "and attempt < 5
"
                        . "order by
priority ")
                                }-> 1 0 ){route };
        die "$date: $0: error: there is the only service:
$route (add
more) \n"
                if ( $route eq "$script $service"
                        or $route eq "$script $service_stored" );
        ( $service name, $service ) = split( /\s+/, $route );
        die "$date: $0: error: no route for the service:
$service (add
more) \n"
                unless $service;
          return $service;
}       __
```

Table 4 below contains source code of the content fetcher **402** used with the content extraction agent **400** to retrieve information from a web site.

TABLE 4

```
#! /usr/local/www/bin/sybperls
#-T
#    -w
#    $Header:
/usr/local/cvsroot/webley/agents/service/webget .pl,v 1.4
# Agent to get info from the web.
# Parameters: service_name Eservice_parameters] , i.e. stock
msft or weather
60645
# configuration stored in files service_name.ini
# if this file is absent the configuration is received from
mcServices table
# This script provides autoupdate to datatable if the .ini
file is newer.
$debug = 1;
use URI::URL;
use LWP::User Agent;
use HTTP::Request::Common;
use Vail::VarList;
use Sybase::CTlib;
use HTTP::cookies;
#print "Sybase::CTlib $DB_USR, $DB_PWD, $DB_SRV;";
open( STDERR, ">>$0.log" ) if $debug;
#open( STDERR, ">&STDOUT" );
$log = 'date';
#$response = './url.pl "http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"';
#$response = 'pwd';
#print STDERR "pwd = $response\n";
#$response = 'ls';
#print STDERR "is = $response\n";
chop( $log );
$log .= "pwd=" . 'pwd';
chop( $log );
#$debug2 = 1;
my $service = shift;
$log .= " $service: ". join( ':', @ARGV ) . "\n";
print STDERR $log if $debug;
#$response = './url.pl
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605"';
my @ini = &read_ini ( $service );
chop( @ini );
my $section = "";
do {$section = &process_section( $section ) } while $section;
#$response = './url.pi
"http://cgi.cnn.com/cgi-bin/weather/redirect'izip=60605"';
exit;
sub read_ini {
                my ( $service ) = my @ini = 0;
                # first, try to read file
                $0 =~ m|^ (.*/) [^//]*|;
                $service = $1 . $service;
                if ( open( INI, "$service.ini" ) ) {
```

TABLE 4-continued

```
                    @ini = ( <INI> );
                    return @ini unless ( $DB_SRV);
                    # update datatable
                    my $file_time = time – int( ( -M "$service.ini" )
* 24 *
3600 )
#                   print "time $file_time\n";
                    my $dbh = new Sybase::CTlib $DB_USR, $DB_PWD,
$DB_SRV;
                    unless ( $dbh ) {
                        print STDERR "webget.pl: Cannot connect to
dataserver $DB_SRV:$DB_USR:$DB_PWD\n";
                        return @ini;
                    }
                    my @row_refs = $dbh–>ct_sql( "select lastUpdate
from
mcServices where service = '$service'", undef, 1 );
                    if ( $dbh–>{Rc }== cs_FAIL ) {
                        print STDERR "webget.pl: DB select from
mcServices
failed\n";
                        return @ini;
                    }
                    unless ( defined @row_refs ) {
                        # have to insert
                        my ( @ini_escaped ) = map {
                            ( my $x = $_ ) =~ } s/\'/\'\'/g
                        $dbh–>ct_sql("insert mcServices values(
'$service'
'@ini_escaped', $file_time ) ');
                        if ( $dbh–>{Rc }== CS_FAIL ) {
                            print STDERR "webget.pl: DB insert to
mcServices failed\n";
                        }
                        return @ini;
                    }
#                   print "time $file_time:".$row_refs [ 0 ]–
> {' lastUpdate'
} . " \n"
                    if ( $file_time > $row_refs [ 0 ] –> {'lastUpdate' })
{
                        # have to update
                        my ( @ini_escaped ) = map {
                            ( my $x = $_ ) =~ s/\'/\'\'/g;
                            $x;
                        }@ini;
                        $dbh–>ct_sql( "update mcServices set config =
'@ini_escaped', lastUpdate = $file_time where service =
'$service'" );
                        if ( $dbh–>{RC }== CS_FAIL ) {
                            print STDERR "webget.pl: DB update to
mcServices failed\n";
                        }
                    }
                    return @ini;
                }
                else {
                    print STDERR "$0: WARNING: $service.ini n/a in " .
'pwd'
                        . "Try to read DB\n";
                }
                # then try to read datatable
                die "webget.pl: Unable to find service $service\n"
unless ( $DB_SRV
);
                my $dbh = new Sybase::CTlib $DB_USR, $DB_PWD, $DB_SRV;
                die "webget.pl: Cannot connect to dataserver
$DB_SRV:$DB_USR:$DB_PWD\n" unless ( $dbh );
                my @row_refs = $dbh–>ct_sql( "select config from
mcServices where
service = '$service'", undef, 1 );
                die "webget.pl: DB select from mcServices failed\n" if
$dbh–>{RC }
== CS_FAIL;
```

TABLE 4-continued

```
                die "webget.pl: Unable to find service $service\n"
unless ( defined
@row_refs );
                $row_refs [ 0 ]->{'config' }=--- s/\n /\n\r/g;
                @ini = split( /\r/, $row_refs [ 0 ] ->{'config' });
                return @ini;
}    -
##########################################
sub process_section {
                my ( $prev_section ) = my ( $section, $output, $content );
                my %Param;
                my %content;
#               print "#####################################\n";
                foreach ( @ini ) {
#               print;
#               chop;
                s/\s+$//;
                s/^\s+//;
                # get section name
                if( /~\[(.* )\]/) {
#                       print "$_: $section:$prev_section\n";
                        last if $section;
                        next if $1 eq "print";
#                       next if $prev_section ne " and $prev_section
ne$1;
                        if ( $prev_section eq $1 ) {
                              $prev_section = "";
                              next;
                        }
                        $section $1;
                    }
                    # get parameters
                    push( @{$Param{$1 }}, $2 ) if $section and
/([^=]+)=(.*)/;
                }
#               print"++++++++++++++++++++++++++++++++++\n";
                return 0 unless $section;
#               print "section $section\n";
                # substitute parameters with values
                map {$Param{URL }->[ 0 ] =- s/$Param{Input }->[ $__
]/$ARGV[ $__
]/g
                }0 .. $#{$Param{Input }};
                # get page content
                ( $content{'TIME' }, $content ) = &geturlcontent(
${$Param{URL
}}[ 0] );
                # filter it
                map {
                    if ( /\" ( [^\"] +)\"([^\"]*)\"/ or
/\/([^\/]+)\/([^\/*)\// )
{
                        my $out = $2; $content =--- s/$1/$out/g;
                    }
                }@{$Param{"Pre-filter" }};
#print STDERR $content;
                # do main regular expression
                unless( @values = $content =~
/${$Param{Regular_expression }} [ 0
]/ ) {
                    &die_hard( ${$Param{Regular_expression }} [ 0 ],
$content
);
                    return $section;
                }
                %content = map {( $Param(Output }->[ $_ ], $values[ $__
])
                }0 .. $*{$Param {Output }};
                # filter it
                map {
                    if ( / ( [^\"] +) \" ([^\"] +) \"([^\ "]*) \"/
                        or / ( [^\/] +) \/ ( [^\/] +) \/ ( ( ^\/] *) \// ) {
                        my $out = $3;
                        $content{$1 }=~ s/$2/$out/g;
                    }
                }@{$Param{"Post-filter" }};
                # calculate it
                map
                    if( /([^=]+)=(.*)/ ) {
                        my $eval = $2;
```

TABLE 4-continued

```
                    map {$eval =-. s/$/$Content{$ }/g
                    }keys %Content;
                $content{$1 }= eval( $eval );
            }
        }@{$Param{Calculate }};
        # read section [print]
        foreach $i ( 0 . . $*ini ) {
            next unless $ini[ $i ] =~ /^\[.+\]/;
            foreach ( $i + 1 . . $#ini ) {
                last if $ini[ $_ ]=~ /^\[.+\]/;
                $output .= $ini[ $_ ] . "\n";
            }
            last;
        }
        # prepare output
        map {$output =~ s/$_/$Content{$_ }/g
        }keys %Content;
        print $output;
        return 0;
}
##############################################
sub get_url content {
        my ( $url ) = @_;
        print STDERR $url if $debug;
#       $response = `./url.pl '$url'`;
        $response = `./url.pl '$url'`;
        return( $time - time, $response );
        my $ua = LWP::UserAgent->new;
        $ua->agent( 'Mozilla/4.0 [en] (X11; I; FreeBSD 2.2.8-
STABLE i386)'
);
#       ua->proxy( ['http', 'https'],
'http://proxy.webley:3128/' );
#       $ua->no_proxy( 'webley', 'vail' );
        my $cookie = HTTP::Cookies->new;
        $ua->cookie_jar( $cookie );
        $url = url $url;
        print "$unl\n" if $debug2;
        my $time = time;
        my $res = $ua->request( GET $uri );
        print "Response: " . ( time - $time ) . "sec\n" if
$debug2;
        return( $time - time, $res->content );
}
##############################################
sub die_hard {
        my( $re, $content ) =
        my ( $re_end, $pattern );
        while( $content /$re/ ) {
            if ( $re =- s/ (\ ( [^\ (\) ] +\) [^\ (\)] *$)// ) }
                $re end = $1 . $re end;
            }
            else {
                $re_end = $re;
                last;
            }
        }
        $content =~ /$re/;
        print STDERR "The regular expression did not match:\n
$ re\n
Possible misuse:
$re_end: \n
Matched:
$&\n
Mismatched:
$'\n
" if $debug;
            if ( $debug) {
                print STDERR "Content:\n $content\n" unless
$';
            }
}
##############################################
```

Table 5 below contains the content descriptor file source code for obtaining weather information from the web site www.cnn.com that is used by the extraction agent **400** of the preferred embodiment.

TABLE 5

```
[cnn]
Input=_zip
URL=http: //cgi .cnn.com/cgi-bin/weather/redinect?zip= zip
Pre-filter="\n"
Pre-filter="<[^<>]+>""
Pre-filter=/\s+/ /
Pre-filter=" (\()\|]"!"
Output=_location
Output=first_day_name
Output=first_day_weather
Output=first_day_high_F
Output=first_day_high_c
Output=first_day_low_F
Output=first_day_low_c
Output=second_day_name
Output=second_day_weather
Output=second_day_high_F
Output=second_day_high_c
Output=second_day_low_F
Output=second_day_low_c
Output=third_day_name
Output=third_day_weather
Output=third_day_high_F
Output=third_day_high_c
Output=thind_day_low_F
Output=thind_day_low_c
Output=fourth_day_name
Output=fourth_day_weather
Output=fourth_day_high_F
Output=fourth_day_high_c
Output=fourth_day_low_F
Output=fourth_day_low_c
Output=undef
Output=_cunrent_time
Output=_current_month
Output=_current_day
Output=_current_weather
Output=_current_tempenature_F
Output=_current_temperature_c
Output=_humidity
Output=_wind
Output=_pressure
Output=_sunrise
Output=_sunset
Regular expression=Author   (.+) Four Day Forecast
(\S+) (\S+) HIGH
(\s+) F (\S+) C LOW (\S+) F (\S+) C (\S+) (\S+) HIGH (\S+) F
(\S+) C LOW
(\S+) F (\S+) C (\S+) (\S+) HIGH (\S+) F (\S+) C LOW (\S+) F
(\S+) C (\S+)
(\S+) HIGH (\S+) F (\S+) C LOW (\S+) F (\S+) C (.+) Current
Conditions(.+)
!local!, (\S+) (\S+) (.+) Temp: (\S+) F, (\S+) C Rel.
Humidity:    (\S+) Wind:
(.+) Pressure: (.+) Sunrise: (.+) Sunset: (.+) Related Links
Post-filter=_current_weather"p/"partly"
Post-filter=_current_weather" l/"little"
Post-filter=_current_weather"m/"mostly"
Post-filter=_current_weathen"t-/"thunder"
Post-filter=_wind"N"North"
Post-filter=_wind"E"East"
Post-filter=_wind"s"South"
Post-filter=_wind"W"West"
Post-filter=_wind/mph/miles per hour/
Post-filter=_wind/kph! /kilometers per hour/
Post-filter=_wind"\s+!","
[print]
Current weather in _location is _current_weather.
Temperature is _current_temperature_F Fahrenheit,
_current_temperature_C
Celsius.
Humidity is _humidity.
Wind from the _wind.
```

Table 6 below contains the content descriptor file source code for obtaining weather information from the web site www.lycos.com that is used by the extraction agent **400** of the preferred embodiment.

TABLE 6

```
[lycos]
Input=zip
Input=_city
URL=http://weather.lycos.com/wcfiveday.asp?city=zip
Pre-filter="\n"
Pne-filter="</TD>"td"
Pre-filter="<! .*?_>""
Pre-filter="<br>" br_"
Pre-filter=/alt"/>alt=/
Pre-filter="<[^<>]+>""
Pre-filter=" "
Pre-filter=/\s+/ /
Output=_location
Output=_current_weather
Output=_current_temperature_F
Output=_humidity
Output=_winddir
Output=_windspeed
Output=_windmeasure
Output=_pressure
Output=finst_day_name
Output=second_day_name
Output=thind_day_name
Output=fourth_day_name
Output=fifth_day_name
Output=first_day_weather
Output=second_day_weather
Output=third_day_weather
Output=founth_day_weather
Output=fifth_day_weather
Output=first_day_high_F
Output=first_day_low_F
Output=second_day_high_F
Output=second_day_low_F
Output=thind_day_high_F
Output=third_day_low_F
Output =fourth_day_high_F
Output=fourth_day_low_F
Output=fifth_day_high_F
Output=fifth_day_low_F
Output=_windkmh
Regular expression=Guide My Lycos (.+) Click image to
enlarge
alt=([^"]+)"(?:.+) Temp: (\d+)(?:.+)F_br_Humidity:
(\S+) (?:.+) Wind:
_br__
Output =_current_temperature_C
Post-filter=_location"_br_"
Post-filter=_current_weather"p/"partly"
Post-filter=_current_weather"m/"mostly"
Post-filter=_current_weather"t-/"thunder"
Post-filter=_winddir"@" at"
Post-filter=_winddir/mph/miles pen hour/
Post-filter=_wind/kph!/kilometens per hour/
Calculate=_current_temperature_C=int ( (_current_temperature_F
-32) *5/9)
Calculate=_windkmh=int (_windspeed*1.6)
[print]
The current weather in _location is _current_weather.
The current temperature is _current_temperature_F Fanenheit
_curnent_temperatune_C Celsius.
Humidity is _humidity.
Winds _winddir.
```

Once the web browsing server **102** accesses the web site specified in the URL **204** and retrieves the requested information, the information is forwarded to the media server **106**. The media server uses the speech synthesis engine **302** to create an audio message that is then transmitted to the user's voice enabled device **112**. In the preferred embodiment, each web browsing server **102** is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

Referring to FIG. **1**, the operation of the robust voice browser system will be described. A user establishes a connection between his voice enabled device **112** and a media server **106**. This may be done using the Public Switched Telephone Network (PSTN) **116** by calling a telephone number associated with the voice browsing system **118**. Once the connection is established, the media server **106** initiates an interactive voice response (WR) application **304**. The IVR application plays audio messages to the user presenting a list of options, such as, "stock quotes", "flight status", "yellow pages", "weather", and "news". These options are based upon the available web site categories and may be modified as desired. The user selects the desired option by speaking the name of the option into the voice enabled device **112**.

As an example, if a user wishes to obtain restaurant information, he may speak into his telephone the phrase "yellow pages". The FIR application would then ask the user what he would like to find and the user may respond by stating "restaurants". The user may then be provided with further options related to searching for the desired restaurant. For instance, the user may be provided with the following restaurant options, "Mexican Restaurants", "Italian Restaurants", or "American Restaurants". The user then speaks into the telephone **112** the restaurant type of interest. The IVR application running on the media server **106** may also request additional information limiting the geographic scope of the restaurants to be reported to the user. For instance, the IVR application may ask the user to identify the zip code of the area where the restaurant should be located. The media server **106** uses the speech recognition engine **300** to interpret the speech commands received from the user. Based upon these commands, the media server **106** retrieves the appropriate web site record **200** from the database **100**. This record and any additional data, which may include other necessary parameters needed to perform the user's request, are transmitted to a web browsing server **102**. A firewall **104** may be provided that separates the web browsing server **102** from the database **100** and media server **106**. The firewall provides protection to the media server and database by preventing unauthorized access in the event the firewall for web browsing server **108** fails or is compromised. Any type of firewall protection technique commonly known to one skilled in the art could be used, including packet filter, proxy server, application gateway, or circuit-level gateway techniques.

The web browsing server **102** then uses the web site record and any additional data and executes the extraction agent **400** and relevant content descriptor file **406** to retrieve the requested information.

The information received from the responding web site **114** is then processed by the web browsing server **102** according to the content descriptor file **406** retrieval by the extraction agent. This processed response is then transmitted to the media server **106** for conversion into audio messages using either the speech synthesis software **302** or selecting among a database of prerecorded voice responses contained within the database **100**.

As mentioned above, each web site record contains a rank number **202** as shown in FIG. **2**. For each category searchable by a user, the database **100** may list several web sites, each with a different rank number **202**. As an example, three different web sites may be listed as searchable under the category of "restaurants". Each of those web sites will be assigned a rank number such as 1, 2, or 3. The site with the highest rank (i.e., rank=1) will be the first web site accessed by a web browsing server **102**. If the information requested by the user cannot be found at this first web site, then the web browsing server **102** will search the second ranked web site and so forth down the line until the requested information is retrieved or no more web sites left to check.

The web site ranking method and system of the present invention provides robustness to the voice browser system and enables it to adapt to changes that may occur as web sites evolve. For instance, the information required by a web site **114** to perform a search or the format of the reported response data may change. Without the ability to adequately monitor and detect these changes, a search requested by a user may provide an incomplete response, no response, or an error. Such useless responses may result from incomplete data being provided to the web site **114** or the web browsing server **102** being unable to recognize the response data messages received from the searched web site **114**.

The robustness and reliability of the voice browsing system of the present invention is further improved by the addition of a polling mechanism. This polling mechanism continually polls or "pings" each of the sites listed in the database **100**. During this polling function, a web browsing server **102** sends brief data requests or "polling digital data" to each web site listed in database **100**. The web browsing server **102** monitors the response received from each web site and determines whether it is a complete response and whether the response is in the expected format specified by the content descriptor file **406** used by the extraction agent **400**. The polled web sites that provide complete responses in the format expected by the extraction agent **400** have their ranking established based on their "response lime". That is, web sites with faster response times will be will be assigned higher rankings than those with slower response times. If the web browsing server **102** receives no response from the polled web site or if the response received is not in the expected format, then the rank of that web site is lowered. Additionally, the web browsing server contains a warning mechanism that generates a warning message or alarm for the system administrator indicating that the specified web site has been modified or is not responsive and requires further review.

Since the web browsing servers **102** access web sites based upon their ranking number, only those web sites that produce useful and error-free responses will be used by the voice browser system to gather information requested by the user. Further, since the ranking numbers are also based upon the speed of a web site in providing responses, only the most time efficient sites are accessed. This system assures that users will get complete, timely, and relevant responses to their requests. Without this feature, users may be provided with information that is not relevant to their request or may not get any information at all. The constant polling and re-ranking of the web sites used within each category allows the voice browser of the present invention to operate efficiently. Finally, it allows the voice browser system of the present invention to dynamically adapt to changes in the rapidly evolving web sites that exist on the Internet.

It should be noted that the web sites accessible by the voice browser of the preferred embodiment may use any type of mark-up language, including Extensible Markup Language (XML), Wireless Markup Language (WML), Handheld Device Markup Language (HDML), Hyper Text Markup Language (HTML), or any variation of these languages.

A second embodiment of the present invention is depicted in FIG. **5**. This embodiment provides a system and method for controlling a variety of devices **500** connected to a network **502** by using conversational speech commands spoken into a voice enabled device **504** (i.e., wireline or wireless telephones, Internet Protocol (IP) phones, or other special wireless units). The networked devices may include various household devices. For instance, voice commands may be

used to control household security systems, VCRs, TVs, outdoor or indoor lighting, sprinklers, or heating and air conditioning systems.

Each of these devices **500** is connected to a network **502**. These devices **500** may contain embedded microprocessors or may be connected to other computer equipment that allow the device **500** to communicate with network **502**. In the preferred embodiment, the devices **500** appear as "web sites" connected to the network **502**. This allows a network interface system, such as a device browsing server **506**, a database **508**, and a user interface system, such as a media server **510**, to operate similar to the web browsing server **102**, database **100** and media server **106** described in the first preferred embodiment above. A network **502** interfaces with one or more network interface systems, which are shown as device browsing servers **506** in FIG. **5**. The device browsing servers perform many of the same functions and operate in much the same way as the web browsing servers **102** discuss above in the first preferred embodiment. The device browsing servers **506** are also connected to a database **508**.

Database **508** lists all devices that are connected to the network **502**. For each device **500**, the database **508** contains a record similar to that shown in FIG. **2**. Each record will contain at least a device identifier, which may be in the form of a URL, and a command to "content extraction agent" contained in the device browsing server **506**. Database **508** may also include any other data or software necessary to test and administer the device browsing system.

The content extraction agent operates similarly to that described in the first embodiment. A device descriptor file contains a listing of the options and functions available for each of the devices **500** connected on the network **502**. Furthermore, the device descriptor file contains the information necessary to properly communicate with the networked devices **500**. Such information would include, for example, communication protocols, message formatting requirements, and required operating parameters.

The device browsing server **506** receives messages from the various networked devices **500**, appropriately formats those messages and transmits them to one or more media servers **510** which are part of the device browsing system. The user's voice enabled devices **504** can access the device browsing system by calling into a media server **510** via the Public Switched Telephone Network (PSTN) **512**. In the preferred embodiment, the device browsing server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The media servers **510** act as user interface systems and perform the functions of natural speech recognition, speech synthesis, data processing, and call handling. The media server **510** operates similarly to the media server **106** depicted in FIG. **3**. When data is received from the device browser server **506**, the media server **510** will convert the data into audio messages via a speech synthesis engine that are then transmitted to the voice enabled device of the user **504**. Speech commands received from the voice enabled device of the user **504** are converted into data messages via a speech recognition engine running on the media server **510**. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www.nuance.com). A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhsl.com). The media servers **510** of the preferred embodiment are based on Intel's Dual Pentium III 730 MHz microprocessor system. A specific example for using the system and method of this embodiment of the invention will now be given.

First, a user may call into a media server **510** by dialing a telephone number associated with an established device browsing system. Once the user is connected, the IVR application of the media server **510** will provide the user with a list of available systems that may be monitored or controlled based upon information contained in database **508**.

For example, the user may be provided with the option to select "Home Systems" or "Office Systems". The user may then speak the command "access home systems". The media server **510** would then access the database **508** and provide the user with a listing of the home subsystems or devices **500** available on the network **502** for the user to monitor and control. For instance, the user may be given a listing of subsystems such as "Outdoor Lighting System", "Indoor Lighting System", "Security System", or "Heating and Air Conditioning System". The user may then select the indoor lighting subsystem by speaking the command "Indoor Lighting System". The IVR application would then provide the user with a set of options related to the indoor lighting system. For instance the media server **510** may then provide a listing such as "Dining Room", "Living Room", "Kitchen", or "Bedroom". After selecting the desired room, the IVR application would provide the user with the options to hear the "status" of the lighting in that room or to "turn on", "turn off", or "dim" the lighting in the desired room. These commands are provided by the user by speaking the desired command into the users voice enabled device **504**. The media server **510** receives this command and translates it into a data message. This data message is then forwarded to the device browsing server **506** which routes the message to the appropriate device **500**.

The device browsing system **514** of this embodiment of the present invention also provides the same robustness and reliability features described in the first embodiment. The device browsing system **514** has the ability to detect whether new devices have been added to the system or whether current devices are out-of-service. This robustness is achieved by periodically polling or "pinging" all devices **500** listed in database **508**. The device browsing server **506** periodically polls each device **500** and monitors the response. If the device browsing server **506** receives a recognized and expected response from the polled device, then the device is categorized as being recognized and in-service. However, if the device browsing server **506** does not receive a response from the polled device **500** or receives an unexpected response, then the device **500** is marked as being either new or out-of-service. A warning message or a report may then be generated for the user indicating that a new device has been detected or that an existing device is experiencing trouble.

Therefore, this embodiment allows users to remotely monitor and control any devices that are connected to a network, such as devices within a home or office. Furthermore, no special telecommunications equipment is required for users to remotely access the device browser system. Users may use any type of voice enabled device (i.e., wireline or wireless telephones, IP phones, or other wireless units) available to them. Furthermore, a user may perform these functions from anywhere without having to subscribe to additional services. Therefore, no additional expenses are incurred by the user.

The descriptions of the preferred embodiments described above are set forth for illustrative purposes and are not intended to limit the present invention in any manner. Equivalent approaches are intended to be included within the scope of the present invention. While the present invention has been described with reference to the particular embodiments illustrated, those skilled in the art will recognize that many

changes and variations may be made thereto without departing from the spirit and scope of the present invention. These embodiments and obvious variations thereof are contemplated as falling within the scope and spirit of the claimed invention.

What is claimed is:

1. A method for retrieving information from web sites by uttering speech commands into a voice enabled device and for providing to users retrieved information in an audio form via said voice enabled device, said method comprising the steps of:

providing a computer operatively connected to the internet, said computer further being operatively connected to at least one speaker-independent speech recognition engine and to at least one speech synthesis engine;

providing a voice enabled device operatively connected to said computer, said voice enabled device configured to receive speech commands from users;

providing a speech command to said speaker-independent speech recognition engine,

said computer accessing at least one of a plurality of web sites associated with said speech command to obtain an information to be retrieved, said computer first accessing a first web site of said plurality of web sites and, if said information to be retrieved is not found at said first web site, said computer sequentially accessing said plurality of web sites until said information to be retrieved is found or until said plurality of web sites has been accessed;

said speech synthesis engine producing an audio message containing any retrieved information from said web sites; and

said speech synthesis engine transmitting said audio message to said users via said voice enabled device.

2. The method of claim **1** wherein said speech command is further associated with a content descriptor associated with each said web site address, said content descriptor pre-defining a portion of said web site containing said information to be retrieved.

3. The method of claim **1** wherein said speech command is further associated with a ranking from highest to lowest associated with each said web site, said ranking indicating the order in which the plurality of web sites are accessed.

4. The method of claim **3** wherein said computer accesses said plurality of web sites based on said ranking, said computer first accessing said web site having the highest ranking.

5. The method of claim **4** further comprising the step of adjusting said rankings associated with said plurality of web sites such that said web site having said information to be retrieved is assigned the highest ranking and any web sites not having said information to be retrieved are assigned lower rankings.

6. The method of claim **1** further comprising the step of periodically polling each said web site to determine whether said web site contains said information to be retrieved.

7. The method of claim **6** wherein the computer periodically polls each said web site without being instructed by said user to determine the availability of each said web site, the duration of time for each said web site to respond to a request from said computer, and changes to the location of said information to be retrieved from each said web site, said computer creating a ranking of said plurality of web sites based on said periodic polling.

8. The method of claim **1** further comprising the step of periodically searching said internet to find new web sites containing said information to be retrieved, and adding said new web sites to said plurality of web sites.

9. A system for retrieving information from web sites by uttering speech commands into a phone and for providing to users retrieved information in an audio form via said phone, said system comprising:

a computer, said computer operatively connected to the internet and to at least one phone;

at least one speaker-independent speech recognition engine, said speaker-independent speech recognition engine operatively connected to said computer;

at least one speech synthesis engine, said speech synthesis engine operatively connected to said computer;

a database, said database operatively connected to said computer, said database containing a plurality of web site addresses;

a content descriptor associated with each said web site address, said content descriptor pre-defining a portion of said web site containing said information to be retrieved;

a ranking from highest to lowest associated with each said web site address, said ranking indicating the order in which the plurality of web sites are accessed;

said speaker-independent speech recognition engine configured to receive from users via said phone a speech command;

said computer configured to access at least one of said plurality of web sites associated with said speech command to obtain said information to be retrieved, said computer configured to first access said web site having the highest ranking and, if said information to be retrieved is not found at said web site having the highest ranking, said computer configured to subsequently access said plurality of web sites in order of rankings until said information to be retrieved is found or until said plurality of web sites has been accessed;

said computer further configured to establish or adjust said rankings associated with said plurality of web sites such that said web site having said information to be retrieved is assigned the highest ranking and any web sites not having said information to be retrieved are assigned lower rankings;

said speech synthesis engine configured to produce an audio message containing any retrieved information from said web sites, and said speech synthesis engine further configured to transmit said audio message to said users via said phone.

10. The system of claim **9** wherein said phone comprises a standard telephone, a cellular phone, or an IP phone.

11. The system of claim **9** wherein said internet is a local area network.

12. The system of claim **9** wherein said internet is a wide area network.

13. The system of claim **9** wherein said internet is the Internet.

14. The system of claim **9** wherein said computer is configured to establish or adjust said rankings associated with said plurality of web sites when instructed by said user to access said plurality of web sites to retrieve said information.

15. The system of claim **9** wherein said computer is configured to establish or adjust said rankings associated with said plurality of web sites based on periodic polling of each of said web sites without being instructed by said user to determine the availability of each said web site, the duration of time for each said web site to respond to a request from said computer, and changes to the location of said information to be retrieved from each said web site.

* * * * *

Exhibit 4

US009769314B2

(12) **United States Patent**
Kurganov

(10) **Patent No.: US 9,769,314 B2**
(45) **Date of Patent: *Sep. 19, 2017**

(54) **PERSONAL VOICE-BASED INFORMATION RETRIEVAL SYSTEM**

(71) Applicant: **Parus Holdings, Inc.**, Bannockburn, IL (US)

(72) Inventor: **Alexander Kurganov**, Buffalo Grove, IL (US)

(73) Assignee: **Parus Holdings, Inc.**, Bannockburn, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/193,517**

(22) Filed: **Jun. 27, 2016**

(65) **Prior Publication Data**

US 2016/0307583 A1 Oct. 20, 2016

Related U.S. Application Data

(63) Continuation of application No. 12/787,801, filed on May 26, 2010, now Pat. No. 9,377,992, which is a (Continued)

(51) **Int. Cl.**
H04M 3/493 (2006.01)
G06F 3/16 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04M 3/4938*** (2013.01); ***G06F 3/167*** (2013.01); ***G06F 17/30769*** (2013.01);
(Continued)

(58) **Field of Classification Search**
CPC ... G06F 3/167; G06F 17/30861; G10L 17/24; G10L 15/06; G10L 15/26; G10L 15/08;
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

D174,465 S 4/1955 Paxton
3,728,486 A 4/1973 Kraus
(Continued)

FOREIGN PATENT DOCUMENTS

CA 1329852 5/1994
EP 0572544 B1 9/1996
(Continued)

OTHER PUBLICATIONS

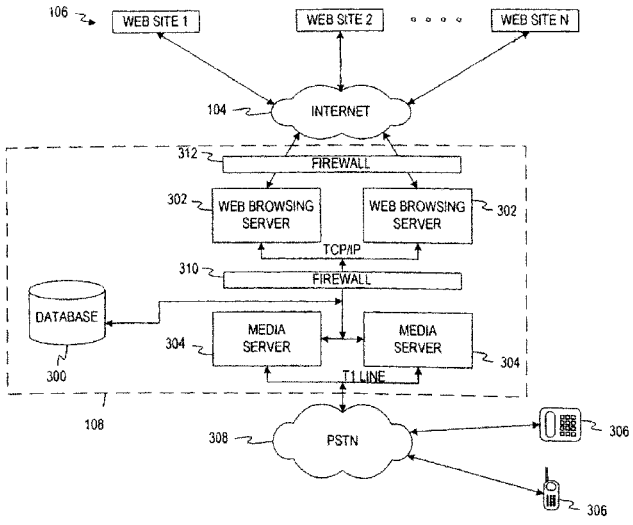
"McGraw-Hill Dictionary of Scientific & Technical Terms 1101, 6th ed. 2003."
(Continued)

Primary Examiner — Lisa Hashem
(74) *Attorney, Agent, or Firm* — Patent Law Works, LLP

(57) **ABSTRACT**

The present invention relates to a system for retrieving information from a network such as the Internet. A user creates a user-defined record in a database that identifies an information source, such as a web site, containing information of interest to the user. This record identifies the location of the information source and also contains a recognition grammar based upon a speech command assigned by the user. Upon receiving the speech command from the user that is described within the recognition grammar, a network interface system accesses the information source and retrieves the information requested by the user.

26 Claims, 5 Drawing Sheets



Related U.S. Application Data

continuation of application No. 11/771,773, filed on Jun. 29, 2007, now abandoned, which is a continuation of application No. 09/777,406, filed on Feb. 6, 2001, now Pat. No. 7,516,190.

(60) Provisional application No. 60/180,343, filed on Feb. 4, 2000.

(51) **Int. Cl.**

G10L 15/26	(2006.01)
G06F 17/30	(2006.01)
G10L 15/08	(2006.01)
G10L 25/54	(2013.01)
G10L 13/08	(2013.01)
G10L 15/02	(2006.01)
G10L 15/22	(2006.01)
G10L 17/24	(2013.01)
G10L 15/06	(2013.01)
H04L 29/08	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 17/30861*** (2013.01); ***G10L 13/08*** (2013.01); ***G10L 15/02*** (2013.01); ***G10L 15/08*** (2013.01); ***G10L 15/22*** (2013.01); ***G10L 15/26*** (2013.01); ***G10L 25/54*** (2013.01); ***H04L 67/02*** (2013.01); *G10L 15/06* (2013.01); *G10L 17/24* (2013.01); *G10L 2015/223* (2013.01); *H04L 29/0809* (2013.01); *H04M 2201/39* (2013.01); *H04M 2201/40* (2013.01); *H04M 2201/405* (2013.01); *H04M 2207/40* (2013.01)

(58) **Field of Classification Search**

CPC G10L 25/54; H04M 2207/40; H04M 2201/405; H04M 3/4938; H04L 29/0809; H04L 67/02

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,058,838	A	11/1977	Crager et al.
4,100,377	A	7/1978	Flanagan
4,158,750	A	6/1979	Sakoe et al.
4,313,035	A	1/1982	Jordan et al.
4,327,251	A	4/1982	Fomenko et al.
4,340,783	A	7/1982	Sugiyama et al.
4,340,797	A	7/1982	Takano et al.
4,340,800	A	7/1982	Ueda et al.
4,371,752	A	2/1983	Matthews et al.
4,481,574	A	11/1984	DeFino et al.
4,489,438	A	12/1984	Hughes
4,500,751	A	2/1985	Darland et al.
4,513,390	A	4/1985	Walter et al.
4,523,055	A	6/1985	Hohl et al.
4,549,047	A	10/1985	Brian et al.
4,584,434	A	4/1986	Hashimoto
4,585,906	A	4/1986	Matthews et al.
4,596,900	A	6/1986	Jackson
4,602,129	A	7/1986	Matthews et al.
4,635,253	A	1/1987	Urui et al.
4,652,700	A	3/1987	Matthews et al.
4,696,028	A	9/1987	Morganstein et al.
4,713,837	A	12/1987	Gordon
4,747,127	A	5/1988	Hansen et al.
4,748,656	A	5/1988	Gibbs et al.
4,755,932	A	7/1988	Diedrich
4,757,525	A	7/1988	Matthews et al.
4,761,807	A	8/1988	Matthews et al.
4,763,317	A	8/1988	Lehman et al.
4,769,719	A	9/1988	Endo
4,771,425	A	9/1988	Baran et al.
4,776,016	A	10/1988	Hansen
4,782,517	A	11/1988	Bernardis et al.
4,792,968	A	12/1988	Katz
4,799,144	A	1/1989	Parruck et al.
4,809,321	A	2/1989	Morganstein et al.
4,811,381	A	3/1989	Woo et al.
4,837,798	A	6/1989	Cohen et al.
4,847,891	A	7/1989	Kotani
4,850,012	A	7/1989	Mehta et al.
4,852,149	A	7/1989	Zwick et al.
4,852,170	A	7/1989	Bordeaux
4,866,758	A	9/1989	Heinzelmann
4,873,719	A	10/1989	Reese
4,879,743	A	11/1989	Burke et al.
4,893,333	A	1/1990	Baran et al.
4,893,335	A	1/1990	Fuller et al.
4,903,289	A	2/1990	Hashimoto
4,903,291	A	2/1990	Tsurufuji et al.
4,905,273	A	2/1990	Gordon et al.
4,907,079	A	3/1990	Turner et al.
4,918,722	A	4/1990	Duehren et al.
4,922,518	A	5/1990	Gordon et al.
4,922,520	A	5/1990	Bernard et al.
4,922,526	A	5/1990	Morganstein et al.
4,926,462	A	5/1990	Ladd et al.
4,930,150	A	5/1990	Katz
4,933,966	A	6/1990	Hird et al.
4,935,955	A	6/1990	Neudorfer
4,935,958	A	6/1990	Morganstein et al.
4,941,170	A	7/1990	Herbst
4,942,598	A	7/1990	Davis
4,953,204	A	8/1990	Cuschleg, Jr. et al.
4,955,047	A	9/1990	Morganstein et al.
4,956,835	A	9/1990	Grover
4,959,854	A	9/1990	Cave et al.
4,967,288	A	10/1990	Mizutori et al.
4,969,184	A	11/1990	Gordon et al.
4,972,462	A	11/1990	Shibata
4,974,254	A	11/1990	Perine et al.
4,975,941	A	12/1990	Morganstein et al.
4,985,913	A	1/1991	Shalom et al.
4,994,926	A	2/1991	Gordon et al.
4,996,704	A	2/1991	Brunson
5,003,575	A	3/1991	Chamberlin et al.
5,003,577	A	3/1991	Ertz et al.
5,008,926	A	4/1991	Misholi
5,020,095	A	5/1991	Morganstein et al.
5,027,384	A	6/1991	Morganstein
5,029,196	A	7/1991	Morganstein
5,036,533	A	7/1991	Carter et al.
5,054,054	A	10/1991	Pessia et al.
5,065,254	A	11/1991	Hishida
5,086,385	A	2/1992	Launey et al.
5,095,445	A	3/1992	Sekiguchi
5,099,509	A	3/1992	Morganstein et al.
5,109,405	A	4/1992	Morganstein
5,128,984	A	7/1992	Katz
5,131,024	A	7/1992	Pugh et al.
5,133,004	A	7/1992	Heileman, Jr. et al.
5,145,452	A	9/1992	Chevalier
5,146,452	A	9/1992	Pekarske
5,166,974	A	11/1992	Morganstein et al.
5,179,585	A	1/1993	MacMillan, Jr. et al.
5,193,110	A	3/1993	Jones et al.
5,195,086	A	3/1993	Baumgartner et al.
5,233,600	A	8/1993	Pekarske
5,243,643	A	9/1993	Sattar et al.
5,243,645	A	9/1993	Bissell et al.
5,249,219	A	9/1993	Morganstein et al.
5,255,305	A	10/1993	Sattar
5,263,084	A	11/1993	Chaput et al.
5,276,729	A	1/1994	Higuchi et al.
5,287,199	A	2/1994	Zoccolillo
5,291,302	A	3/1994	Gordon et al.
5,291,479	A	3/1994	Vaziri et al.
5,303,298	A	4/1994	Morganstein et al.
5,307,399	A	4/1994	Dai et al.
5,309,504	A	5/1994	Morganstein
5,325,421	A	6/1994	Hou et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,327,486 A 7/1994 Wolff et al.
5,327,529 A 7/1994 Fults et al.
5,329,578 A 7/1994 Brennan et al.
5,333,266 A 7/1994 Boaz et al.
5,347,574 A 9/1994 Morganstein
5,355,403 A 10/1994 Richardson, Jr. et al.
5,359,598 A 10/1994 Steagall et al.
5,365,524 A 11/1994 Hiller et al.
5,365,574 A 11/1994 Hunt et al.
5,375,161 A 12/1994 Fuller et al.
5,384,771 A 1/1995 Isidoro et al.
5,404,231 A 4/1995 Bloomfield
5,408,526 A 4/1995 McFarland et al.
5,414,754 A 5/1995 Pugh et al.
5,416,834 A 5/1995 Bales et al.
5,426,421 A 6/1995 Gray
5,432,845 A 7/1995 Burd et al.
5,436,963 A 7/1995 Fitzpatrick et al.
5,459,584 A 10/1995 Gordon et al.
5,463,684 A 10/1995 Morduch et al.
5,475,791 A 12/1995 Shalk et al.
5,479,487 A 12/1995 Hammond
5,495,484 A 2/1996 Self et al.
5,497,373 A 3/1996 Hulen et al.
5,499,288 A 3/1996 Hunt et al.
5,515,427 A 5/1996 Carlsen et al.
5,517,558 A 5/1996 Schalk
5,526,353 A 6/1996 Henley et al.
5,533,115 A 7/1996 Hollenbach et al.
5,537,461 A 7/1996 Bridges et al.
5,555,100 A 9/1996 Bloomfield et al.
5,559,611 A 9/1996 Bloomfield et al.
5,559,859 A 9/1996 Dai et al.
5,566,236 A 10/1996 MeLampy et al.
5,603,031 A 2/1997 White et al.
5,608,786 A 3/1997 Gordon
5,610,910 A 3/1997 Focsaneanu et al.
5,610,970 A 3/1997 Fuller et al.
5,611,031 A 3/1997 Hertzfeld et al.
5,630,079 A 5/1997 McLaughlin
5,652,789 A 7/1997 Miner et al.
5,657,376 A 8/1997 Espeut et al.
5,659,597 A 8/1997 Bareis et al.
5,666,401 A 9/1997 Morganstein et al.
5,675,507 A 10/1997 Bobo, II
5,675,811 A 10/1997 Broedner et al.
5,689,669 A 11/1997 Lynch et al.
5,692,187 A 11/1997 Goldman et al.
5,699,486 A 12/1997 Tullis et al.
5,712,903 A 1/1998 Bartholomew et al.
5,719,921 A 2/1998 Vysotsky et al.
5,721,908 A 2/1998 Lagarde et al.
5,724,408 A 3/1998 Morganstein
5,737,395 A 4/1998 Irribarren
5,742,596 A 4/1998 Baratz et al.
5,742,905 A 4/1998 Pepe et al.
5,752,191 A 5/1998 Fuller et al.
5,758,322 A 5/1998 Rongley
5,761,294 A 6/1998 Shaffer et al.
5,764,639 A 6/1998 Staples et al.
5,764,736 A 6/1998 Shachar et al.
5,764,910 A 6/1998 Shachar
5,774,860 A 6/1998 Bayya et al.
5,787,298 A 7/1998 Broedner et al.
5,793,993 A 8/1998 Broedner et al.
5,794,205 A 8/1998 Walters et al.
5,796,791 A 8/1998 Polcyn
5,799,063 A 8/1998 Krane
5,799,065 A 8/1998 Junqua et al.
5,809,282 A 9/1998 Cooper et al.
5,809,481 A 9/1998 Baron et al.
5,812,796 A 9/1998 Broedner et al.
5,819,220 A 10/1998 Sarukkai et al.
5,819,306 A 10/1998 Goldman et al.
5,822,727 A 10/1998 Garberg et al.
5,823,879 A 10/1998 Goldberg et al.
5,832,063 A 11/1998 Vysotsky et al.
5,832,440 A 11/1998 Woodbridge
5,835,570 A 11/1998 Wattenbarger
5,838,682 A 11/1998 Dekelbaum et al.
5,867,494 A 2/1999 Krishnaswamy et al.
5,867,495 A 2/1999 Elliott et al.
5,870,550 A 2/1999 Wesinger, Jr. et al.
5,873,080 A 2/1999 Coden et al.
5,881,134 A 3/1999 Foster et al.
5,881,135 A 3/1999 Watts et al.
5,884,032 A 3/1999 Bateman et al.
5,884,262 A 3/1999 Wise et al.
5,884,266 A 3/1999 Dvorak
5,890,123 A 3/1999 Brown et al.
5,905,476 A 5/1999 McLaughlin et al.
5,914,951 A 6/1999 Bentley et al.
5,915,001 A 6/1999 Uppaluru
5,917,817 A 6/1999 Dunn et al.
5,926,789 A 7/1999 Barbara et al.
5,940,598 A 8/1999 Strauss et al.
5,943,399 A 8/1999 Bannister et al.
5,946,389 A 8/1999 Dold
5,953,392 A * 9/1999 Rhie H04M 1/2478 379/88.13
5,974,124 A 10/1999 Schlueter et al.
5,974,413 A 10/1999 Beauregard et al.
5,991,292 A 11/1999 Focsaneau et al.
5,995,615 A 11/1999 Miloslavsky
5,999,525 A 12/1999 Krishnaswamy et al.
5,999,611 A 12/1999 Tatchell et al.
5,999,965 A 12/1999 Kelly
6,012,088 A 1/2000 Li et al.
6,014,437 A 1/2000 Acker et al.
6,014,626 A 1/2000 Cohen
6,018,710 A 1/2000 Wynblatt et al.
6,021,181 A 2/2000 Miner et al.
6,021,190 A 2/2000 Fuller et al.
6,031,904 A 2/2000 An et al.
6,038,305 A 3/2000 McAllister et al.
6,044,107 A 3/2000 Gatherer et al.
6,047,053 A 4/2000 Miner et al.
6,052,372 A 4/2000 Gittins et al.
6,067,516 A 5/2000 Levay et al.
6,078,580 A 6/2000 Mandalia et al.
6,081,518 A 6/2000 Bowman-Amuah
6,081,782 A 6/2000 Rabin
6,091,808 A 7/2000 Wood et al.
6,101,472 A 8/2000 Giangarra et al.
6,104,803 A 8/2000 Weser et al.
6,115,737 A 9/2000 Ely et al.
6,115,742 A 9/2000 Franklin et al.
6,130,933 A 10/2000 Milsoslavsky
6,131,095 A 10/2000 Low et al.
6,137,863 A 10/2000 Brown et al.
6,144,991 A 11/2000 England
6,157,705 A 12/2000 Perrone
6,161,128 A 12/2000 Smyk
6,178,399 B1 1/2001 Takebayashi et al.
6,185,535 B1 2/2001 Hedin et al.
6,188,683 B1 2/2001 Lang et al.
6,195,357 B1 2/2001 Polcyn
6,199,076 B1 3/2001 Logan et al.
6,201,814 B1 3/2001 Greenspan
6,201,863 B1 3/2001 Miloslavsky
6,208,638 B1 3/2001 Rieley et al.
6,215,858 B1 4/2001 Bartholomew et al.
6,230,132 B1 5/2001 Class et al.
6,233,318 B1 5/2001 Picard et al.
6,243,373 B1 6/2001 Turock
6,252,944 B1 6/2001 Hansen, II et al.
6,269,336 B1 7/2001 Ladd et al.
6,285,745 B1 9/2001 Bartholomew et al.
6,327,572 B1 12/2001 Morton et al.
6,330,538 B1 12/2001 Breen
6,343,529 B1 2/2002 Pool
6,349,132 B1 2/2002 Wesemann et al.
6,353,661 B1 3/2002 Bailey, III
6,366,575 B1 4/2002 Barkan et al.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,366,578 B1 4/2002 Johnson
6,424,945 B1 7/2002 Sorsa
6,430,282 B1 8/2002 Bannister et al.
6,434,529 B1 8/2002 Walker et al.
6,445,694 B1 9/2002 Swartz
6,446,076 B1 9/2002 Burkey et al.
6,456,699 B1 9/2002 Burg
6,459,910 B1 10/2002 Houston
6,477,240 B1 11/2002 Lim et al.
6,477,420 B1 11/2002 Struble et al.
6,490,627 B1 12/2002 Kalra et al.
6,501,966 B1 12/2002 Bareis et al.
6,505,163 B1 1/2003 Zhang et al.
6,529,948 B1 3/2003 Bowman-Amuah
6,532,444 B1 3/2003 Weber
6,539,359 B1 3/2003 Ladd et al.
6,546,393 B1 4/2003 Khan
6,560,604 B1 5/2003 Fascenda
6,584,439 B1 6/2003 Geilhufe et al.
6,587,822 B2 7/2003 Brown et al.
6,593,944 B1 7/2003 Nicolas et al.
6,594,348 B1 7/2003 Bjurstrom et al.
6,594,692 B1 7/2003 Reisman
6,606,611 B1 8/2003 Khan
6,618,039 B1 9/2003 Grant et al.
6,618,726 B1 9/2003 Colbath et al.
6,618,763 B1 9/2003 Steinberg
6,636,831 B1 10/2003 Profit, Jr. et al.
6,654,814 B1 11/2003 Britton et al.
6,658,662 B1 12/2003 Nielsen
6,665,640 B1 12/2003 Bennett et al.
6,687,341 B1 2/2004 Koch et al.
6,704,024 B2 3/2004 Robotham et al.
6,718,015 B1 4/2004 Berstis
6,721,705 B2 4/2004 Kurganov et al.
6,724,868 B2 4/2004 Pradhan et al.
6,732,142 B1 5/2004 Bates et al.
6,763,388 B1 7/2004 Tsimelzon
6,771,732 B2 8/2004 Xiao et al.
6,771,743 B1 8/2004 Butler et al.
6,775,264 B1 8/2004 Kurganov
6,785,266 B2 8/2004 Swartz
6,807,257 B1 10/2004 Kurganov
6,812,939 B1 11/2004 Flores et al.
6,823,370 B1 11/2004 Kredo et al.
6,888,929 B1 5/2005 Saylor et al.
6,922,733 B1 7/2005 Kuiken et al.
6,941,273 B1 9/2005 Loghmani et al.
6,964,012 B1 11/2005 Zirngibl et al.
6,964,023 B2 11/2005 Maes et al.
6,965,864 B1 11/2005 Thrift et al.
6,996,609 B2 2/2006 Hickman et al.
6,999,804 B2 2/2006 Engstrom et al.
7,003,463 B1 2/2006 Maes et al.
7,024,464 B1 4/2006 Lusher et al.
7,050,977 B1 5/2006 Bennett
7,075,555 B1 7/2006 Flores
7,076,431 B2 7/2006 Kurganov et al.
7,089,307 B2 8/2006 Zintel et al.
7,145,898 B1 12/2006 Elliott
7,146,323 B2 12/2006 Guenther et al.
7,327,723 B2 2/2008 Kurganov
7,386,455 B2 6/2008 Kurganov
7,506,022 B2 3/2009 Wang et al.
7,516,190 B2 4/2009 Kurganov
7,881,941 B2 2/2011 Kurganov et al.
7,974,875 B1 7/2011 Quilici et al.
8,098,600 B2 1/2012 Kurganov
8,131,267 B2 3/2012 Lichorowic et al.
8,131,555 B1 3/2012 Carriere et al.
8,185,402 B2 5/2012 Kurganov et al.
8,380,505 B2 2/2013 Konig et al.
8,775,176 B2 7/2014 Gilbert et al.
8,838,074 B2 9/2014 Kurganov
8,843,120 B2 9/2014 Kurganov
8,843,141 B2 9/2014 Kurganov
8,874,446 B2 10/2014 Carriere et al.
9,451,084 B2 9/2016 Kurganov et al.
2001/0011302 A1 8/2001 Son
2001/0032234 A1 10/2001 Summers et al.
2001/0040885 A1 11/2001 Jonas et al.
2001/0048676 A1 12/2001 Jimenez et al.
2002/0006126 A1 1/2002 Johnson et al.
2002/0059402 A1 5/2002 Belanger
2002/0064149 A1 5/2002 Elliott et al.
2002/0087327 A1 7/2002 Lee et al.
2002/0090114 A1 7/2002 Rhoads et al.
2002/0104025 A1 8/2002 Wrench
2003/0002635 A1 1/2003 Koch et al.
2004/0160913 A1 8/2004 Kubler et al.
2004/0247094 A1 12/2004 Crockett
2005/0025133 A1 2/2005 Swartz
2005/0030179 A1 2/2005 Script et al.
2005/0074104 A1 4/2005 Swartz
2005/0102147 A1 5/2005 Ullrich et al.
2005/0278179 A1 12/2005 Overend
2006/0069926 A1 3/2006 Ginter et al.
2007/0136072 A1 6/2007 Sampath
2007/0206737 A1 9/2007 Hickman
2007/0249406 A1 10/2007 Andreasson
2007/0263601 A1 11/2007 Kurganov
2007/0286360 A1 12/2007 Chu et al.
2008/0228494 A1 9/2008 Cross
2009/0286514 A1 11/2009 Lichorowic
2010/0042413 A1* 2/2010 Simpson G10L 15/30 704/270
2010/0094635 A1* 4/2010 Bermudez Perez G06F 17/30861 704/270.1
2011/0035220 A1* 2/2011 Opaluch G10L 15/265 704/246
2011/0054898 A1 3/2011 Phillips et al.
2011/0082696 A1* 4/2011 Johnston G06F 17/30026 704/243
2011/0091023 A1 4/2011 Kurganov et al.
2011/0153324 A1* 6/2011 Ballinger G10L 15/30 704/235
2012/0179464 A1 7/2012 Newman
2012/0253800 A1 10/2012 Goller
2013/0006638 A1 1/2013 Lindahl
2013/0041666 A1* 2/2013 Bak G10L 15/26 704/246
2013/0191122 A1 7/2013 Mason
2013/0317823 A1 11/2013 Mengibar
2014/0039898 A1 2/2014 Reich
2014/0046660 A1 2/2014 Kamdar
2014/0111415 A1 4/2014 Gargi et al.
2014/0123010 A1 5/2014 Goldstein
2015/0134340 A1* 5/2015 Blaisch G06F 3/04842 704/275
2015/0185985 A1 7/2015 Kang et al.
2015/0234636 A1 8/2015 Barnes, Jr.
2015/0334080 A1 11/2015 Tamayo
2015/0339745 A1 11/2015 Peter et al.
2016/0057383 A1* 2/2016 Pattan H04M 3/4938 348/14.03
2016/0080811 A1* 3/2016 Fukushima G06Q 30/06 725/59
2016/0125881 A1* 5/2016 Vogel G06F 17/30746 704/9
2016/0179752 A1* 6/2016 Clark G06F 3/167 704/270.1
2016/0225369 A1* 8/2016 Agrawal G06F 3/0484
2016/0239497 A1* 8/2016 O'Donnell G06F 17/30867
2016/0321266 A1* 11/2016 Philippov G06F 3/167
2016/0328206 A1* 11/2016 Nakaoka G06F 3/16
2017/0116986 A1* 4/2017 Weng G10L 15/22

FOREIGN PATENT DOCUMENTS

EP 0794650 A2 9/1997
GB 2211698 7/1989
GB 2240693 A 8/1991

(56) **References Cited**

FOREIGN PATENT DOCUMENTS

GB	2317782 A	4/1998
JP	1-258526	10/1989
WO	91/07838	5/1991
WO	91/18466	11/1991
WO	9609710	3/1996
WO	97/34401	9/1997
WO	97/37481	9/1997
WO	98/23058	5/1998

OTHER PUBLICATIONS

Newton, Harry, Newtons Telecom Dictionary—The Official Glossary of Telecommunications and Voice Processing Terms, Dec. 1992, 6 pages.

Paper No. 10, Denying Institution of Covered Business Method Patent Review CBM2015-00109 and CBM2015-00149, Nov. 9, 2015, 19 pages.

Paper No. 10, Denying Institution of Covered Business Method Patent Review CBM2015-00110 and CBM2015-00150, Nov. 9, 2015, 20 pages.

Paper No. 10, Denying Institution of Covered Business Method Patent Review CBM2015-00111 and CBM2015-00151, Nov. 9, 2015, 19 pages.

Paper No. 10, Denying Institution of Covered Business Method Patent Review CBM2015-00112 and CBM2015-00152, Nov. 9, 2015, 18 pages.

Putz, Steve, Interactive Information Services Using World-Wide Web Hypertext, First Int'! Conference on World-Wide Web (May 25-27,1994), 10 pages.

Memorandum Opinion and Order, Oct. 8, 2015, 27 pages.

Update Subject Matter Eligibility, Jul. 2015, 33 pages.

Wikipedia Definition of "Internet", available at http://en.wikipedia.org/wiki/Internet pp. 24-26.

Opening Brief of Appellant Paws Holdings, Inc., submitted on Mar. 8, 2016, to the United States Court of Appeal for the Federal Circuit, 236 pages.

Brief of Appellees, submitted on Jun. 20, 2016, to the United States Court of Appeals for the Federal Circuit, 53 pages.

"A PABX that Listens and Talks", Speech Technology, Jan./Feb. 1984, pp. 74-79.

Amended Complaint, *Parus Holdings. Inc.* v. *Web Telephony LLC & Robert Swartz*, Case No. 06-cv-O 1146 (N.D. III.), Jul. 10, 2006, 14 pages.

AT&T, Press Release, "AT&T Customers Can Teach Systems to Listen and Respond to Voice", Jan. 17, 1995, pp. 1-2, Basking Ridge, NJ., available at www.lucent.com/press/O 195/950117.gbb.html (accessed Mar. 15, 2005).

Bellcore Technology Licensing, "The Electronic Receptionist—A Knowledge-Based Approach to Personal Communications", 1994, pp. 1-8.

Brachman et al., "Fragmentation in Store-and-Forward Message Transfer", IEEE Communications Magazine, vol. 26 (7), Jul. 1998, pp. 18-27.

"Business Phone Systems for Advanced Offices", NTT Review, vol. 2 (6), Nov. 1990, pp. 52-54.

Cole et al., "An Architecture for a Mobile OSI Mail Access System", IEEE Journal on Selected Areas in Communications, vol. 7 (2), Feb. 1989, pp. 249-256.

"Data Communications Networks: Message Handling Systems", Fasciele, VIII. ?-Recommendations X.400-X.430, 38 pages, date unknown.

DAX Systems, Inc., Press Release, "Speech Recognition Success in DAX's Grasp", Nov. 22, 1995, pp. 1-2, Pine Brook, NJ.

Defendants Answer to the Amended Complaint and Demand for Jury Trial, *Parus Holdings, Inc.* v. *Web Telephone LLC & Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Aug. 10, 2006, 14 pages.

Faxpak Store and Forward Facsimile Transmission Service, Electrical Communication, vol. 54 (3), 1979, pp. 251-55.

Garcia et al., "Issues in Multimedia Computer-Based Message Systems Design and Standardization", NATO ASI Series, vol. 1-6, 1984, 18 pgs.

"Globecom '85 IEEE Global Telecommunications Conference," New Orleans, LA., Dec. 2-5, 1985, pp. 1295-1300.

Hemphill et al., "Speech-Aware Multimedia," IEEE MultiMedia, Spring 1996, vol. 3, No. 1, pp. 74-78, IEEE. As indicated on the cover page of the journal, which is attached hereto as Attachment 4, the reference was received by Cornell University on Mar. 25, 1996.

Hunt et al., "Long-Distance Remote Control to the Rescue", Chicago Tribune, Jun. 15, 2002, Section 4, p. 15.

"Introducing PIC SuperFax, First PC/Fax System to Run Under Windows", Pacific Image Communications, Pasadena, CA, Date Unknown, (received at COMDEX show, Nov. 3, 1987). 4 pgs.

Kubala et al., "BYBLOS Speech Recognition Benchmark Results", Workshop on Speech &Natural Language, Feb. 19-22, 1991.According to the web site http://portal.acm.org/citation.cfm?id~II2405.I 12415&coll . . . , the reference was published in 1991, Morgan Kaufman Publishers, San Franscisco, CA. The distribution date is not presently known.

Ly, "Chatter: A Conversational Telephone Agent", submitted to Program in Media Arts & Sciences, MIT, 1993, pp. 1-130.

Maeda, et al., "An Intelligent Customer-Controlled Switching System", IEEE Global Telecommunications Conference, Hollywood, Florida, Nov. 28-Dec. 1, 1988, pp. 1499-1503.

Markowitz, J., "The Ultimate Computer Input Device May Be Right Under Your Nose", Byte, Dec. 1995, pp. 1-13, available atwww.byte.com/art/9512/sec8/artl.htm (accessed Mar. 15, 2005).

Marx et al., "Mail Call: Message Presentation and Navigation in a Nonvisual Environment," SIGCHI Conference on Human Factors in Computing Systems, Vancouver, B.C., Canada, Apr. 13-18, 1996. The web site http://www.usabilityviews.corn/uv001673.html shows a date of Apr. 16, 1996. The distribution date is not presently known.

Marx, M., "Toward Effective Conversational Messaging" (Thesis). As indicated on the cover page, the thesis was presented to the Departmental Committee on Graduate Students, Program in Media Arts and Sciences, School of Architecture and Planning, Massachusetts Institute of Technology on May 12, 1995. According to the web site http://www.thesis.mit.edu/Dienst/Repository/2.0/Body/0018.mit.theses/1995-314/rfc1807bib., the thesis was indexed on Mar. 21, 2000.

Oye, Phil, "Juggler", p. 1, availableathttp://www.philove.com/work/juggler/index.shtml (accessed on Dec. 8, 2006).

Oye, Phil, "Juggler", p. 1, available athttp://www.philoye.com/work/juggler_2.shtml (accessed on Dec. 8, 2006).

Perdue et al., "Conversant® 1 Voice System: Architecture and Applications",Jul. 17, 1986, AT&T Technical Journal, pp. 1-14.

Plaintiff Parus Holdings, Inc. 's Supplemental Responses to Defendant Web Telephone LLC's First Set of Interrogatories (Nos. 1-12), *Parus Holdings, Inc.* v. *Web Telephony LLC Y Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Oct. 31, 2006, 32 pages.

Plaintiff Parus Holdings, Inc. 's Supplemental Responses to Defendant Web Telephony LLC's Second Set of Interrogatories (Nos. 13-17), *Parus Holdings,Inc.* v. *Web Telephony LLC & Robert Swartz*, Case No. 06-cv-01146 (N.D. III.), Oct. 31, 2006, 31 pages.

Print outs of Internet web site, "Wildfire Communications, Inc.,", Nov. 5, 1997, including print outs of the following web pages: http://www.wildfire.com; http://www.wildfire.com/consumerhome.html;http://www.wildfire.com/106.html; http://www.wildfire.com/carrierhome.html; http://www.wildfire.com/sfandb.html; http://www.wildfire.com/about.html; http://www.wildfire.com/abtmgmt.html; http://www. wildfire.corn/scoop.html; http://www. wildfire.corn/intel.html; and http://www.wildfire.com/msft.html.

"Proceedings of the IFIP 10th World Computer Congress", Dublin, Ireland, Sep. 1-5, 1986.

"PureSpeech Announces Juggler PC System for First Quarter of 1997", HighBeam Research, Sep. 19, 1996, pp. 1-3, available at http://www.highbeam.com/doc/1G1-186909545.html (accessed on Dec. 8, 2006).

PureSpeech, "Meet the Voice of Juggler!", pp. 1-3, the date of Nov. 18, 1996 is shown at the top of p. 1.

"PureSpeech's Juggler", Teleconnect, Dec. 1996 issue, p. 36.

(56) **References Cited**

OTHER PUBLICATIONS

Ross, Randy, "Retrieve E-mail from a Telephone", Oct. 7, 1996, pp. 1-2, available at http://resna.org/ProfessOrg?Sigs?SIGSites/sig11archive/juggler.htm (accessed on Dec. 8, 2006). Printout indicates that the article was originally printed in PC world.

Sartori, M., "Speech Recognition", Apr. 1995, pp. 1-9, Mercury Communications, available at www.gar.co.uk/technology_watch/speech.htm (accessed Mar. 15, 2005).

Schmandt et al., "A Conversational Telephone Messaging System", IEEE Transactions on Consumer Electronics, 1984, vol. CE-30, No. 3, pp. xxi-xxiv.

Schmandt et al., "Phone Slave: A Graphical Telecommunications Interface", Proceedings of the SID, 1985, vol. 26/1, pp. 79-82.

"Secretarial Branch Exchange", IBM Technical Disclosure Bulletin, vol. 26 (5), Oct. 1983, pp. 2645-2647.

Shimamura, et al., "Review of the Electrical Communication Laboratories", vol. 418 (33), No. 1, Tokyo, Japan, 1985, pp. 31-39.

"The VMX Systems Product Reference Manual: Product Description Volume", May 1994, vol. 1, release 7.1, VMX, Inc. (Octel Communications Corp.) San Jose, CA USA.

"VMXworks Product Reference Manual: vol. 3 Programmer's Guide", Jul. 1994, vols. 3 & 4, Release 3.1, Octel Communications Corp., Milpitas, CA, USA.

"Wildfire Communication, Inc.", Harvard Business School, Mar. 21, 1996, Pub!. No. 9-396-305, pp. 1-22.

"WordPerfect: New Telephony Features Boost Office", WordPerfect Office TechBrief, 1994, Info-World Publishing. Co., vol. 10, Issue 2, pp. 2-3.

Yang, C., "INET Phone—Telephone Services and Servers on the Internet", Apr. 1995, University of North Texas, pp. 1-6.

Bellcore Technology Licensing, "The Electronic Receptionist—A Knowledge-Based Approach to Personal Communications," 1994, pp. 1-8.

Examples: Abstract Ideas, 20 pages.

IBM AIX DirectTalk/6000 Version 1 Release 6 Improves Your Voice Processing Services to Callers and Customers, Announcement No. 295-489, Nov. 28, 1995, 27 pages.

IBM Announcement Letter No. A95-893, retrieved on Mar. 9, 2015, 10 pages.

IBM, AIX DirectTalk/6000: General Information and Planning, Release 6, GC33-I 720-00, Dec. 1995, 162 pages.

IBM, DirectTalkMail: Administration, Release 6, SC33-I 733-00, Feb. 1996, 274 pages.

"McGraw-Hill Dictionary of Scientific & Technical Terms 1101, 6th ed. 2003," No copy is provided but please inform if a copy of this dictionary is required.

Hemphill et al., "Surfing the Web by Voice," ACM Multimedia 95—Electronic Proceedings, Nov. 5-9, 1995, 8 pages, San Francisco, CA.

IBM, AIX DirectTalk/6000 Release 6: Speech Recognition with the BBN Hark Recognizer, SC33-1734-00, Feb. 1996, 250 pages.

Joint Appendix, submitted on Sep. 16, 2016, to the United States Court of Appeal for the Federal Circuit, 406 pages.

Juggler by PureSpeech, p. 1, available at http://members.aol.com/compqanda1/juggler.html (accessed on Dec. 8, 2006).

Reply Brief of Appellant Parus Holdings, Inc., submitted on Sep. 6, 2016, to the United States Court of Appeal for the Federal Circuit, 40 pages.

Judgment without Opinion for *Parus Holdings Inc.*, v. *Sallie Mae Bank*, Navient Solutions Inc., PNC Bank, N.A., Suntrust Bank, Suntrust Mortgage Inc., 2016-1179, 2016-1180, 2016-1181, entered Feb. 27, 2017 (2 pages).

* cited by examiner

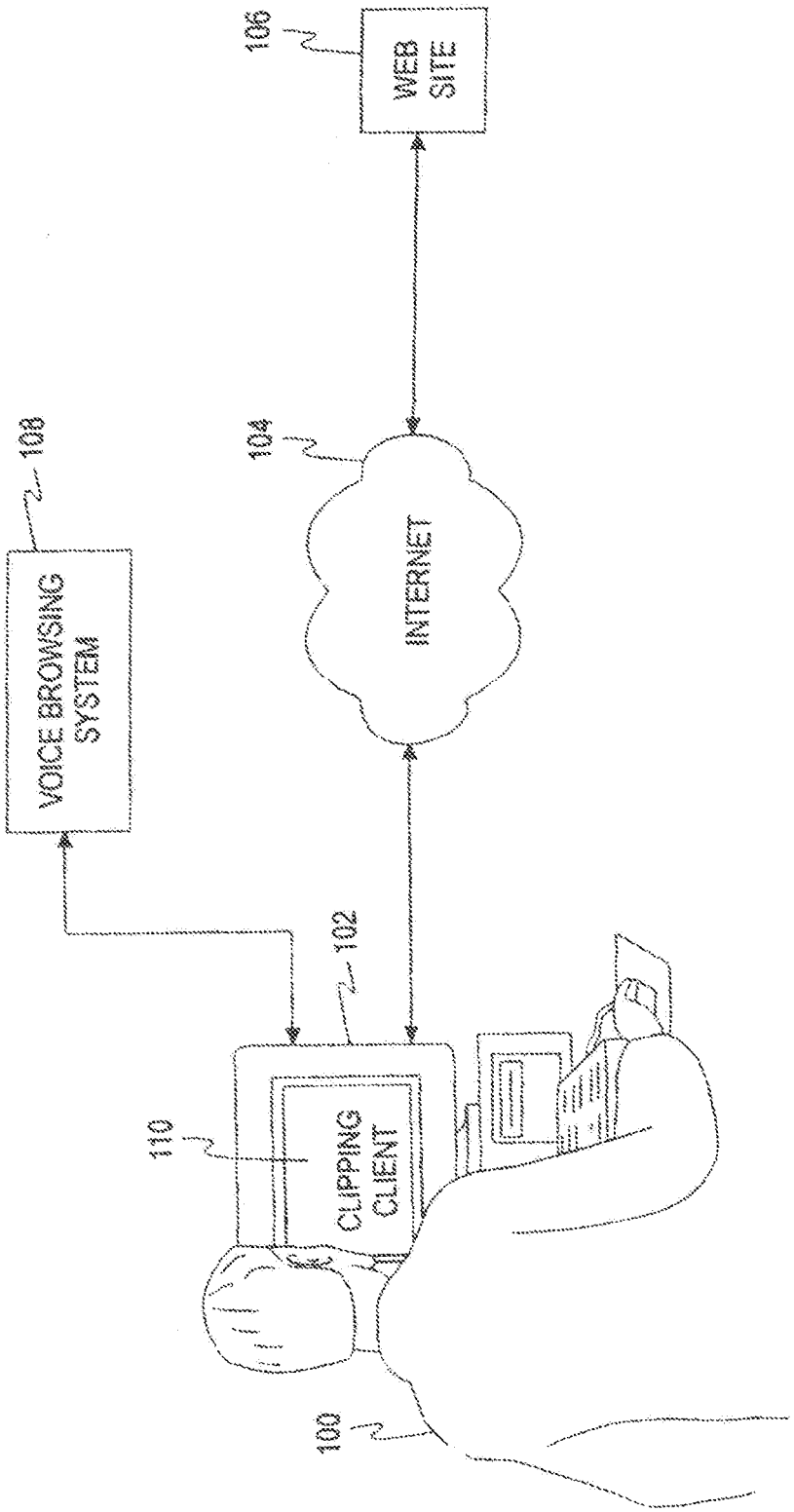

106
WEB SITE
104
INTERNET
108
VOICE BROWSING SYSTEM
102
110
CLIPPING CLIENT
100

FIG. 1

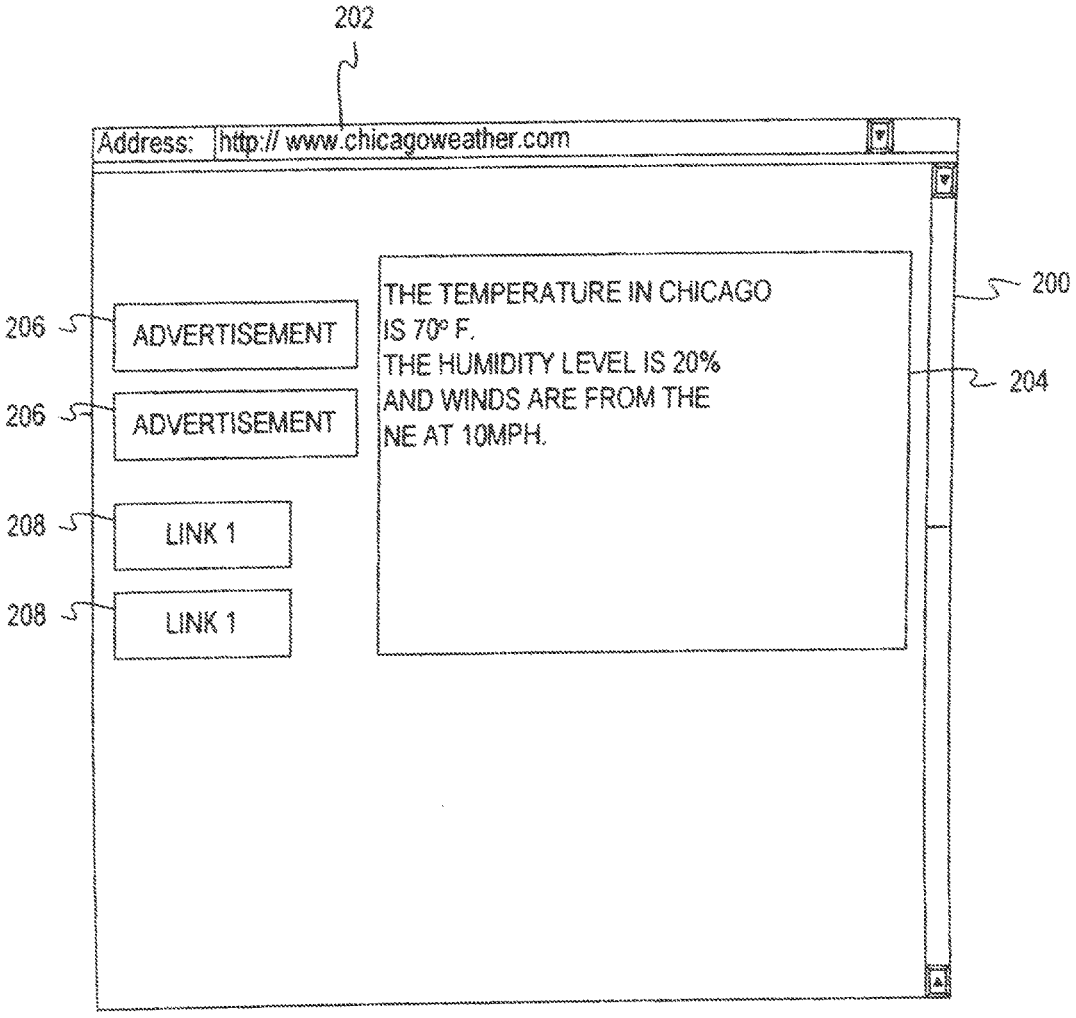

202
Address: http:// www.chicagoweather.com
200
206
ADVERTISEMENT
206
ADVERTISEMENT
THE TEMPERATURE IN CHICAGO
IS 70° F.
THE HUMIDITY LEVEL IS 20%
AND WINDS ARE FROM THE
NE AT 10MPH.
204
208
LINK 1
208
LINK 1

FIG. 2

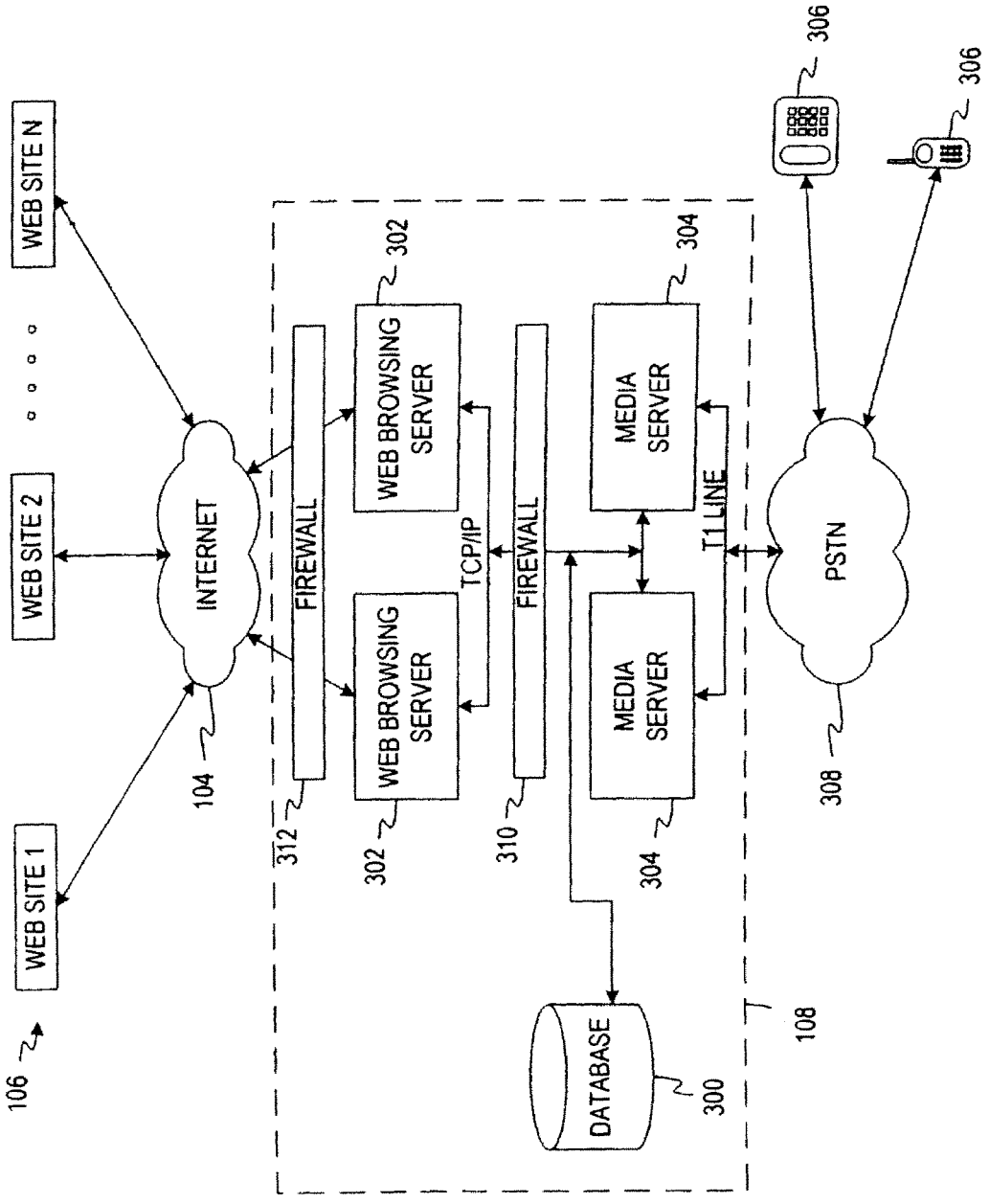

WEB SITE N
WEB SITE 2
WEB SITE 1
106
INTERNET
104
FIREWALL
312
WEB BROWSING SERVER
302
WEB BROWSING SERVER
302
TCP/IP
FIREWALL
310
MEDIA SERVER
304
MEDIA SERVER
304
T1 LINE
PSTN
308
306
306
DATABASE
300
108

FIG. 3

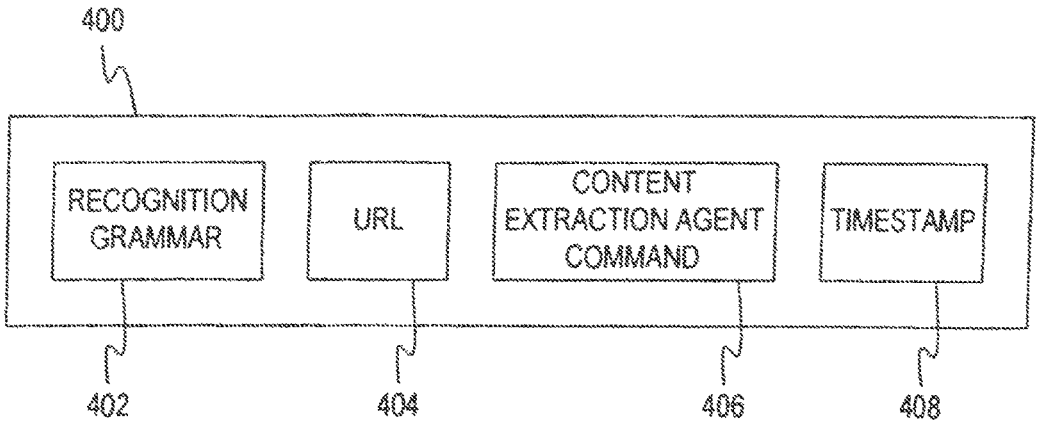

400
RECOGNITION GRAMMAR
URL
CONTENT EXTRACTION AGENT COMMAND
TIMESTAMP
402
404
406
408

FIG. 4

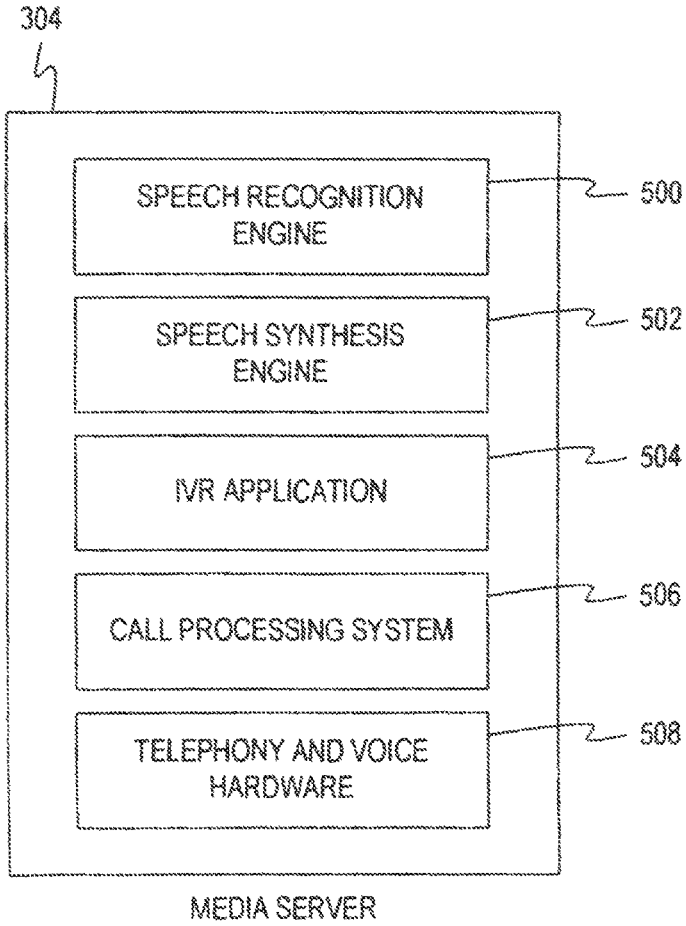

304
SPEECH RECOGNITION ENGINE
500
SPEECH SYNTHESIS ENGINE
502
IVR APPLICATION
504
CALL PROCESSING SYSTEM
506
TELEPHONY AND VOICE HARDWARE
508
MEDIA SERVER

FIG. 5

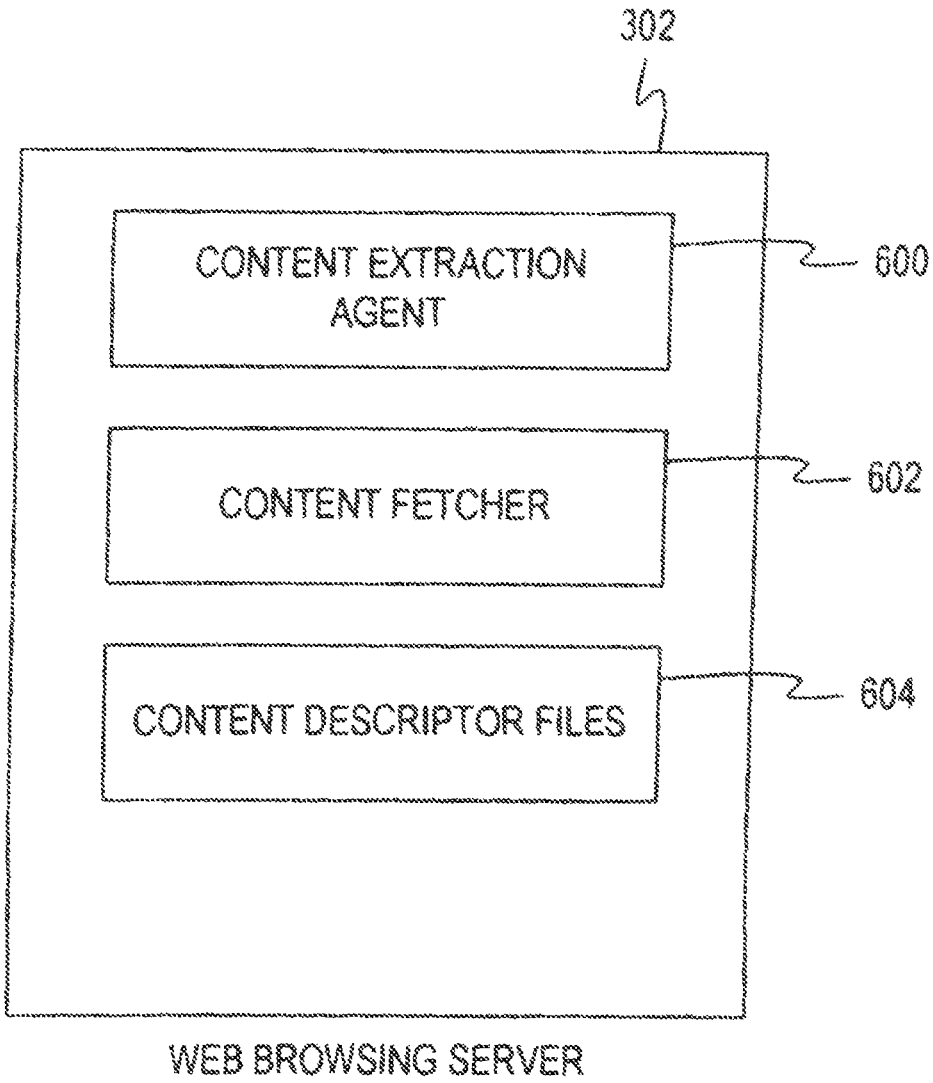

302
CONTENT EXTRACTION AGENT
600
CONTENT FETCHER
602
CONTENT DESCRIPTOR FILES
604
WEB BROWSING SERVER

FIG. 6

PERSONAL VOICE-BASED INFORMATION RETRIEVAL SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is a continuation of U.S. Utility application Ser. No. 12/787,801, filed May 26, 2010, which is a continuation of U.S. Utility application Ser. No. 11/711,773, filed Jun. 29, 2007, which is a continuation of U.S. Utility application Ser. No. 09/777,406, dated Feb. 6, 2001, which claims priority to U.S. Provisional Patent Application No. 60/180,343, filed Feb. 4, 2000, which are incorporated by reference herein in their entirety.

FIELD OF THE INVENTION

The present invention relates generally to the field of providing information IO access. In particular, the invention relates to a personalized system for accessing information from the Internet or other information sources using speech commands.

BACKGROUND OF THE INVENTION

Popular methods of information access and retrieval using the Internet or other computer networks can be time-consuming and complicated. A user must frequently wade through vast amounts of information provided by an information source or web site in order obtain a small amount of relevant information. This can be time-consuming, frustrating, and, depending on the access method, costly. A user is required to continuously identify reliable sources of information and, if these information sources are used frequently, repeatedly access these sources.

Current methods of accessing information stored on computer networks, such as Wide Area Networks (WANs), Local Area Network (LANs) or the Internet, require a user to have access to a computer While computers are becoming increasingly smaller and easier to transport, using a computer to access information is still more difficult than simply using a telephone. Since speech recognition systems allow a user to convert his voice into a computer-usable message, telephone access to digital information is becoming more and more feasible Voice recognition technology is growing in its ability to allow users to use a wide vocabulary.

Therefore, a need exists for an information access and retrieval system and method that allows users to access frequently needed information from information sources on networks by using a telephone and simple speech commands.

SUMMARY OF THE INVENTION

One object of the preferred embodiment of the present invention is to allow users to customize a voice browsing system.

A further object of the preferred embodiment is to allow users to customize the information retrieved from the Internet or other computer networks and accessed by speech commands over telephones.

Another object of the preferred embodiment is to provide a secure and reliable retrieval of information over the Internet or other computer networks using predefined verbal commands assigned by a user.

The present invention provides a solution to these and other problems by providing a new system for retrieving

information from a network such as the Internet. A user creates a user-defined record in a database that identifies an information source, such as a web site, containing information of interest to the user. This record identifies the location of the information source and also contains a recognition grammar assigned by the user. Upon receiving a speech command from the user that is described in the assigned recognition grammar, a network interface system accesses the information source and retrieves the information requested by the user.

In accordance with the preferred embodiment of the present invention, a customized, voice-activated information access system is provided. A user creates a descriptor file defining specific information found on a web site the user would like to access in the future. The user then assigns a pronounceable name or identifier to the selected content and this pronounceable name is saved in a user-defined database record as a recognition grammar along with the URL of the selected web site.

In the preferred embodiment, when a user wishes to retrieve the previously defined web-based information, a telephone call is placed to a media server. The user provides speech commands to the media server that are described in the recognition grammar assigned to the desired search. Based upon the recognition grammar, the media server retrieves the user-defined record from a database and passes the information to a web browsing server which retrieves the information from associated web site. The retrieved information is then transmitted to the user using a speech synthesis software engine.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** displays a personal information selection system used with the preferred embodiment of the present invention;

FIG. **2** displays a web page displayed by the clipping client of the preferred embodiment;

FIG. **3** is a block diagram of a voice browsing system used with preferred embodiment of the present invention;

FIG. **4** is a block diagram of a user-defined database record created by preferred embodiment of the present invention;

FIG. **5** is a block diagram of a media server used by the preferred embodiment; and

FIG. **6** is a block diagram of a web browsing server used by the preferred embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention uses various forms of signal and data transmission to allow a user to retrieve customized information from a network using speech communication. In the preferred embodiment of the present invention, a user associates information of interest found on a specific information source, such as a web site, with a pronounceable name or identification word. This pronounceable name/identification word forms a recognition grammar in the preferred embodiment. When the user wishes to retrieve the selected information, he may use a telephone or other voice enabled device to access a voice browser system. The user then speaks a command described in the recognition grammar associated with the desired information. The voice browsing system then accesses the associated information source and returns to the user, using a voice synthesizer, the requested information.

Referring to FIG. **1**, a user **100** uses a computer **102** to access a network, such as a WAN, LAN, or the Internet, containing various information sources [n the preferred embodiment, the user **100** access the Internet **104** and begins searching for web sites **106**, which are information sources that contain information of interest to the user. When the user **100** identifies a web site **106** containing information the user would like to access using only a voice enabled device, such as a telephone, and the voice browsing system **108**, the user initiates a "clipping client" engine **110** on his computer **102**.

The clipping client **110** allows a user **100** to create a set of instructions for use by the voice browsing system **108** in order to report personalized information back to the user upon request. The instruction set is created by "clipping" information from the identified web site. A user **100** may be interested in weather for a specific city, such as Chicago. The user **100** identifies a web site from which he would like to obtain the latest Chicago weather information. The clipping client **110** is then activated by the user **100**.

The clipping client **110** displays the selected web site in the same manner as a conventional web browser such as Microsoft's® Internet Explorer. FIG. **2** depicts a sample of a web page **200** displayed by the clipping client **110**. The user **100** begins creation of the instruction set for retrieving information from the identified web site by selecting the uniform resource locator (URL) address **202** for the web site (i.e., the web site address). In the preferred embodiment, this selection is done by highlighting and copying the URL address **202**. Next, the user selects the information from the displayed web page that he would like to have retrieved when a request is made. Referring to FIG. **2**, the user would select the information regarding the weather conditions in Chicago **204**. The web page **200** may also contain additional information such as advertisements **206** or links to other web sites **208** which are not of interest to the user. The clipping client **110** allows the user to select only that portion of the web page containing information of interest to the user. Therefore, unless the advertisements **206** and links **208** displayed on the web page are of interest to the user, he would not select this information. Based on the web page information **204** selected by the user, the clipping client **110** creates a content descriptor file containing a description of the content of the selected web page. This content descriptor file indicates where the information selected by the user is located on the web page. In the preferred embodiment, the content descriptor file is stored within the web browsing server **302** shown in FIG. **3**. The web browsing server **302** will be discussed below.

Table 1 below is an example of a content descriptor file created by the clipping client of the preferred embodiment. This content descriptor file relates to obtaining weather information from the web site www.cnn.com.

TABLE 1

```
table name : portalServices
column :
  service
content:
  weather
column:
  config
content:
  [cnn]
  Input=__zip
URL=http://cgi.cnn.com/cgi-bin/weather/redirect?zip= zip
Pre-filter="\n" "
```


TABLE 1-continued

```
Pre-filter = " < [ " <: > ] + > " "
Pre-filter=/\s+/ I
Pre-filter=" [ \ ( \) \ I ] " ! "
Output=__location
Output=first__day__name
Output=first__day__weather
Output=first__day__high__F
Output=first__day__high__C
Output=first__day__low__F
Output=first__day__low__c
Output=second__day__name
Output=second__day__weather
Output=second__day__high__F
Output=second__day__high__C
Output=second__day__low__F
Output=second__day__low__C
Output=third__day__name
Output=third__day__weather
Output=third__day__high__F
Output=third__day__high__C
Output=third__day__low__F
Output=third__day__low__C
Output=fourth__day__name
Output=fourth__day__weather
Output=fourth__day__high__F
Output=fourth__day__high__C
Output=fourth__day__low__F
Output=fourth__day__low__C
Output=undef
Output=__current__time
Output=fourth__day__low__C
Output=undef
Output=__current__time
Output=__current__month
Output=__current__day
Output=__current__weather
Output=__current__temperature__F
Output=__current__temperature__C
Output=__humidity
Output=__wind
Output=__pressure
Output=__sunrise
Output=__sunset
  Regular__expression=WEB SERVICES: (.+) Forecast FOUR-DAY
  FORECAST (\S+)
(\S+) HIGH
(\S+) F (\S+) C LOW (\S+) F (\S+) C (\S+) (\S+) HIGH (\S+) F
(\S+) C LOW
(\S+
) F (\S+) C (\S+) (\S+) HIGH (\S+) F (\S+) C LOW (\S+) F
(\S+) C (\S+) (\S+)
HIGH
-(\S+) C LOW (\S+) F (\S+) C WEATHER MAPS RADAR ( .+) Forecast
CURRENT CONDITIONS
(.+) !local!, (\S+) (\S+) (.+) Temp: (\S+) F,
(\S+) C Rel.
Humidity: (
\S+) Wind: (.+) Pressure: ( .+) Sunrise: ( .+) Sunset: ( .+)
```

Finally, the clipping client **110** prompts the user to enter an identification word or phrase that will be associated with the identified web site and information. For example, the user could associate the phrase "Chicago weather" with the selected URL **202** and related weather information **204**. The identification word or phrase is stored as a personal recognition grammar that can now be recognized by a speech recognition engine of the voice browsing system **108** which will be discussed below. The personal recognition grammar, URL address **202**, and a command for executing a content extraction agent are stored within a database used by the voice browser system **108** which will be discussed below.

The voice browsing system **108** used with the preferred embodiment will now be described in relation to FIG. **3**. A database **300** designed by Webley Systems Incorporated is connected to one or more web browsing servers **302** as well as to one or more media servers **304**. The database may store

information on magnetic media, such as a hard disk drive, or it may store information via other widely acceptable methods for storing data, such as optical disks. The media servers **304** function as user interface systems that provide access to the voice browsing system **108** from a user's voice enabled device **306** (i.e., any type of wireline or wireless telephone, Internet Protocol (IP) phones, or other special wireless units). The database **300** contains a section that stores the personal recognition grammars and related web site information generated by the clipping client **110**. A separate record exists for each web site defined by the user. An example of a user-defined web site record is shown in FIG. **4**. Each user-defined web site record **400** contains the recognition grammar **402** assigned by the user, the associated Uniform Resource Locator (URL) **404**, and a command that enables the "content extraction agent" **406** and retrieves the appropriate content descriptor file required to generate proper requests to the web site and to properly format received data. The web-site record **400** also contains the timestamp **408** indicating the last time the web site was accessed. The content exaction agent is described in more detail below.

The database **300** may also contain a listing of pre-recorded audio files used to create concatenated phrases and sentences. Further, database **300** may contain customer profile information, system activity reports, and any other data or software servers necessary for the testing or administration of the voice browsing system **108**.

The operation of the media servers **304** will now be discussed in relation to FIG. **5**. The media servers **304** function as user interface systems since they allow a user to access the voice browsing system **108** via a voice enabled device **306**. In the preferred embodiment, the media servers **304** contain a speech recognition engine **500**, a speech synthesis engine **502**, an Interactive Voice Response (IVR) application **504**, a call processing system **506**, and telephony and voice hardware **508** that is required to enable the voice browsing system **108** to communicate with the Public Switched Telephone Network (PSTN) **308**. In the preferred embodiment, each media server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

The speech recognition function is performed by a speech recognition engine **500** that converts voice commands received from the user's voice enabled device **10** (i.e., any type of wire line or wireless telephone, Internet Protocol (IP) phones, or other special wireless units) into data messages. In the preferred embodiment voice commands and audio messages are transmitted using the PSTN **308** and data is transmitted using the TCP/IP communications protocol. However, one skilled in the art would recognize that other transmission protocols may be used. Other possible transmission protocols would include SIP/VoIP (Session Initiation Protocol/Voice over IP), Asynchronous Transfer Mode (ATM) and Frame Relay. A preferred speech recognition engine is developed by Nuance Communications of 1380 Willow Road, Menlo Park, Calif. 94025 (www nuance.com) The Nuance engine capacity is measured in recognition units based on CPU type as defined in the vendor specification The natural speech recognition grammars (i.e., what a user can say that will be recognized by the speech recognition engine) were developed by Webley Systems.

In the preferred embodiment, when a user access the voice browsing system **108**, he will be prompted if he would like to use his "user-defined searches." If the user answers affirmatively, the media servers **304** will retrieve from the database **300** the personal recognition grammars **402** defined by the user while using the clipping client **10**.

The media servers **304** also contain a speech synthesis engine **502** that converts the data retrieved by the web browsing servers **302** into audio messages that are transmitted to the user's voice enabled device **306**. A preferred speech synthesis engine is developed by Lernout and Hauspie Speech Products, 52 Third Avenue, Burlington, Mass. 01803 (www.lhslcom).

A further description of the web browsing server **302** will be provided in relation to FIG. **6**. The web browsing servers **302** provide access to data stored on any computer network including the Internet **104**, WANs or LANs. The web browsing servers **302** receive responses from web sites **106** and extract the data requested by the user. This task is known as "content extraction." The web browsing server **302** is comprised of a content extraction agent **600**, a content fetcher **602**, and the content descriptor file **604**. Each of these are software applications and will be discussed below.

Upon receiving a user-defined web site record **400** from the database **300** in response to a user request, the web browsing server **302** invokes the "content extraction agent" command **406** contained in the record **400**. The content extraction agent **600** retrieves the content descriptor file **604** associated with the user-defined record **400**. As mentioned, the content descriptor file **604** directs the extraction agent where to extract data from the accessed web page and how to format a response to the user utilizing that data. For example, the content descriptor file **604** for a web page providing weather information would indicate where to insert the "city" name or ZIP code in order to retrieve Chicago weather information. Additionally, the content descriptor file **604** for each supported URL indicates the location on the web page where the response information is provided. The extraction agent **600** uses this information to properly extract from the web page the information requested by the user.

The content extraction agent **600** can also parse the content of a web page in which the user-desired information has changed location or format. This is accomplished based on the characteristic that most hypertext documents include named objects like tables, buttons, and forms that contain textual content of interest to a user. When changes to a web page occur, a named object may be moved within a document, but it still exists. Therefore, the content extraction agent **600** simply searches for the relevant name of desired object. In this way, the information requested by the user may still be found and reported regardless of changes that have occurred.

Table 2 below contains source code for a content extraction agent **600** used by the preferred embodiment.

TABLE 2

```
# ! /usr/ local/www/bin/sybperl5
#$Header:
/usr/local/cvsroot/webley/agents/service/web__dispatch.pl,v
1.6
# Dispatches all web requests
#http://wcorp.itn.net/cgi/flstat?carrier=ua&flight__no=155&mcn
__abbr=jul&date=
6&stamp=ChLN~PdbuuE*itn/ord,itn/cb/sprint__hd
#http://cig.cnnfn.cm/flightview/rlm?airline=amt&number=300
require "config__tmp.pl";
# check parameters
die "Usage: $0 service [params]\n" if $#ARGV < 1;
#print STDERR @ARGV;
# get parameters
my ( $service, @param ) = @ARGV;
# check service
My ($services = (
```

TABLE 2-continued

```
                    weather_cnn => 'webget.pl weather_cnn',
                    weather_lycos => 'webget.pl
'weather_lycos',
                    weather_weather => 'webget.pl
weather_weather',
                    weather_snap => 'webget.pl
weather_snap',
                    weather_infospace => 'webget.pl
weather_infospace',
                    stockQuote_yahoo => 'webget.pl stock',
                    flightStatus_itn => 'webget.pl
flight_delay',
                    yellowPages_yahoo => 'yp_data.pl',
                    yellowPages_yahoo => 'yp_data.pl',
                    newsHeaders_newsreal => 'news.pl',
                    newsArticle_newsreal => 'news.pl',
                    ) ;
# test param
my $date= 'date';
chop ( $date );
my ( $short_date ) = $date = ~ / \s+({w3}\s+\d{1, 2}) \s+/;
my %Test = (
                    weather_cnn => '60053',
                    weather_lycos => '60053',
                    weather_weather => '60053',
                    weather_snap => '60053',
                    weather_infospace => '60053',
                    stockQuote_yahoo => 'msft',
                    flightStatus_itn => 'ua 155 ' .
$short_date,
                    yellowPages_yahoo => 'tires 60015',
                    newsHeaders_newsreal => ' 1 ',
                    newsArticle_newsreal => '1 1' ,
                    ) ;
die "$date: $0: error: no such service: $service (check
this script) \n"
unless $Services{ $service };
# prepare absolute path to run other scripts
my ( $path, $script ) = $0 =~ m^ (.*/)([ ^/ ] * ) | ;
# store the service to compare against datatable
my $service_stored = $service;
# run service
while( ! ( $response = '$path$Services { $service } @param' )
) (
        # response failed
        # check with test parameters
        $response = '$path$Services { $service } $Test{
$service }",
        If ( $response ) {
        $service = &switch_service( $service ) ;
#       print "wrong parameter values were supplied;
$service -
@param\n";
#       die "$date: $0: error: wrong parameters: $service
-
@param\n";
    }
    else {
        # change priority and notify
        $service = &increase_attempt( $service ) ;
    }
}
# output the response
print $response;
sub increase_attempt {
        my ( $service ) = @_;
        my ( $service_name ) split( /_/, $service ) ;
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                .'set priority = ( select max ( priority
) from
mcServiceRoute "
                . "where service = '$service name' ) + 1,
                . "date = getdate( ), "
                . "attempt = attempt + 1 "
                . "where route = '$script $service' " ) ;
#       print "---$route===\n";
```

TABLE 2-continued

```
        # find new route
        my $route @{ &db_query( "select route from
mcServiceRoute "
                .'where service =
'$service_name' "
                .'and attempt < 5
"
                . "order by
priority ")
                } -> [ 0 ]{ route };
        &db_query( "update mcServiceRoute "
            . "set attempt = 0 "
            . "where route = '$script $service' " ) ;
          if ( $route eq "$script $service_stored" ) ;
        ( $service_name, $service ) =split( /\s+/, $route ) ;
        die "$date: $0: error: no route for the service:
$service (add
More) \n""
            unless $service;
        return $service;
}
sub switch service {
        my ( $service ) = @_;
        my ( $service_name ) = split( /_/, $service );
        print STDERR "$date: $0: attn: changing priority for
service:
$service\n";
        # update priority
        &db_query( "update mcServiceRoute "
                . "set priority = ( select max( priority for
) from
mcServiceRoute "
                . "where service = '$service_name' ) + 1,
                . "date ~ getdate ( ) "
                . "where route = '$script $service' " );
#       print "---$route===\n";
    -   # find new route
        my $route = @( &db_query( "select route from
mcServiceRoute "
                    . "where service =
'$service_name' "
                    . "and attempt < 5
"
                    . "order by
priority ")
                } -> [ 0 ] { route };
        die " $ date : $ 0 : error : there is the only service:
$route (add
more) \n"
        if ( $route eq "$script $service"
            or $route eq "$script $service_stored" ) ;
        (service_name, $service ) = split( / \s+/, $route ) ;
        die "$date: $0: error: no route for the service:
$service (add
more)\n"
            unless $service;
        return $service;
}
```

Table 3 below contains source code of the content fetcher **602** used with the content extraction agent **600** to retrieve information from a web site

TABLE 3

```
#!/usr/local/www/bin/sybper15
#-T
# -w
# $Header:
/usr/local/cvsroot/webley/agents/service/webget.pl, v 1.4
# Agent to get info from the web.
# Parameters: service_name [service_parameters], i.e. stock
msft or weather
60645
# Configuration stored in files service_name.ini
# if this file is absent the configuration is received from
mcServices table
# This script provides autoupdate to datatable if the .ini
```

TABLE 3-continued

```
file is newer.
$debug = 1;
use URI : : URL;
use LWP : : UserAgent;
use HTTP : :Request: : Common;
use Vail : :VarList;
use Sybase : : CT lib;
use HTTP: :Cookies;
#print "Sybase: :CT lib $DB_USR, $DB_PWD, $DB SRV;";
Open ( STDERR, ">>$0.log" ) if $debug;
#open ( STDERR, ">&STDOUT" );
$log = 'date';
#$response = './url.pl
http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605";
#$response= 'pwd';
#print STDERR "pwd = $response\n";
#$response = 'ls' ;
#print STDERR "ls = $response\n";
chop ( $log ) ;
$log .= "pwd=" .   'pwd' ;
chop ( $log ) ;
#$debug2 = 1;
my $service = shift;
$log .= " $service: ". join( ' : ', @ARGV ) . "\n";
print STDERR $log if $debug;
#$response = '. /url .pl
"http://cgi.cnn.com/cgi-bin/weather/redirect?zip=60605" ;
my @ini = &read_ini ( $service ) ;
chop ( @ ini ) ;
my $section= " ";
do ($section = &process_section( $section ) } while
$section;
#$response = ' ./url.pl
http://cgi.cnn.com/cgi-bin.weather/redirect?zip=60605" ' ;
exit;
############################################################
sub read_ini {
    my ( $service ) = @_;
    my @ini = ( );
    # first, try to read file
    $0 =~ m|^ ( .*/) [^/]*|;
    $service = $1 . $service;
    if ( open( INI, "$service.ini" ) ) {
        @ini = ( < INI > ) ;
        return @ini unless ( $DB_SRV ) ;
        # update datatable
        my $file_time = time - int ( ( -M "$service.ini" )
* 24 *
3600 ) ;
#       print "time $file_time\n";
        my $dbh = new Sybase: :CTlib $DB_USR, $DB_PWD,
$DB_SRV;
        unless ( $dbh) {
            print STDERR "webget.pl: Cannot connect to
dataserver $DB_SRV:$DB_USR:$DB_PWD\n";
            return @ini;
        }
        my @row_refs = $dbh->ct_sql ( "select lastUpdate
from
mcServices where service = '$service' ", undef, 1 );
        if ( $dbh -> { RC } == CS_FAIL ) {
            print STDERR "webget.pl: DB select from
mcServices
failed\n";
            return @ini;
        }
        unless ( defined @row_refs ) {
        # have to insert
        my ( @ini_escaped ) = map {
            ( my $x = $_) =~ s/ \ ' / \ ' / g;
            $x;
        }@ini;
        $dbh -> ct_sql ( "insert mcServices values (
'$service',
'@ini_escaped', $file time; ) ");
        if ( $dbh -> { RC } = = CS_FAIL )
            print STDERR "webget.pl: DB insert to
mcServic:es failed\n";
        }
        return @ ini;
#       print "time $file_time:"$row_refs [ 0 ] -> {
'lastUpdate'
}."\n";
        If ( $file_time -> ref_refs [0 ] -> { 'last update'
} ) {
        # have to update
    my ( @ini_escaped = map {
        ( my $x = $_ ) =~ s/ \ ' / \ ' \ ' / g;
        $x;
    } @ini;
    $dbh -> ct_sql ( "update mcServices set config
=
'@ini_escaped', lastUpdate = $file_time where service =
'$service' " );
        if ( $dbh -> { RC } - CS_FAIL ) {
            print STDERR "webget.pl: DB update to
mcServices failed\n";
            }
        }
        return @ini;
    }
    else {
    print STDERR "$0: WARNING: $service.ini n/a in "
. - 'pwd'
        . "Try to read DB\n";
    }
    # then try to read datatable
    die "webget.pl: Unable to find service $service\n"
unless ( $DB_SRV
) ;
    my $dbh = new Sybase: : CTlib $DB_USR, $DB_PWD,
    $DB_SRV;
    die "webget.pl: Cannot connect to dataserver
$DB SRV: $08 USR: $08 PWD\n" unless ( $dbh ) ;
my @row_refs = $dbh->ct sql ( "";;elect con.fiJ from
mcServices where
service = '$service' " , undef, 1 );
    die "webget.pl: DB select from mcServices failed\n" if
$dbh -> { RC }
= = CS FAIL;
    die "webget.pl: Unable to find service $service\n"
unless ( defined
@row_refs ) ;
    $row_refs [ 0 ] -> { 'config' } =~ s/\n /\n\r/g;
    @ini = split ( /\r/, $row_refs [ 0 ] ->{ ' config' } ) ;
    return @ini;
############################################################
sub process_section {
    my ($prev_section ) = @_;
    my ( $section, $output, $content );
    my %PAram;
    my %Content;
#   print" ##################################\n";
    foreach (@ini ) {
        print;
        chop;
        s/\s+$//;
        s/^\[(.*) \ ] ) {
        # get section name
        if ( /^\ [(.*) \ ] ) {
#       print "$_: $section:$prev_section\n";
            last if $section;
            next if $1 eq "print";
#       next if $prev_section ne " " and
$prev_section ne $1;
        if ($prev_section eq $1 )
          $prev_section = " ";
          next;
        }
        $section = $1;
    }
    # get parameters
    Push ( @{ $Param{ $1 } }, $2 ) if $section and
/ ( [ ^ = ] +) = (.*) /;
-   }
#   print"++++++++++++++++++++++++++++++++++\n";
    return 0 unless $section;
#     print "section $section\n";
    # substitute parameters with values
    map { $Param{ URL }->[ 0 ] =~ s/$Param{ Input }->[ $_
```

TABLE 3-continued

```
] /$ARGV [ $__
] /g
      }0 . . S# { $Param{ Input } };
      # get page content
      ( $Content{={ 'TIME' }, $content ) = get__url__content (
$ { $ Param { URL
} } [ 0 ] ) ;
      # filter it
      map {
            if (/\"([^\"]+)\"([^\"]*)\"/or
/\/([^\/]+)\/([^\/]*)\//)
(
                  my $out = $2; $content =~ s/$1/$out/g;
            }
      } @ ($Param{ "Pre-filter"}};
#print STDERR $content;
      # do main regular expression
      unless ( @values = $content =~ / $! Param {
Regular expression } } [ 0
} / ) {
      __hard ( $ { $Param(Requlr__expression } } [ 0
], $content
) ;
            return $section;
      }
      %Content = map { ( $Param{ Output }->[ $__] , $values [
$__ ] )
      } 0 . . $ # ( $Param { Output } ) ;
      # filter it
      map {
      if ( / ( [^\"]+)\"] +) \" ( [^\"]+) \" ( [^\"]*) \"/
            or / ( [ \/]+) \/ ( [ \/] +) \/ ([^\/]*) \/ / ) (
            my $out = $3;
            $Content{ $1 } =~ s/$2/$out/g;
      }
}     @{ $Param { "Post-filter" } };
#calculate it
map
# calculate it
map {
            if ( /([^'=]+)=(.*)/
            my $eval = $2;
            map { $eval =~ s/$__/$Content( $__ }/g
            } keys %Content;
      $Content{ $1 } = eval( $eval ) ;
    }
} @{ ( $Param{ Calculate } } ;
# read section [print]
foreach $i ( 0 .. $#ini ) {
      next unless $ini [ $i] /^\ [print\]/;
      foreach ( $i + 1 . . $#ini ) {
            last if $ini [ $__ ] =~ /^\ [.+\]/;
            $output .= $ini [$__1] . "\n";
      }
      last;
}
# prepare output
map { $output =~ s/$__/$Content{ $__ }/g
} keys %Content;
print $output;
return 0;
}
############################################################
sub get__url__content [
      my ( $url ) = @__;
      print STDERR $url if $debug;
      $response = ' ./url.pl '$url' ;
      $response = ' ./url.pl '$url' ;
      Return( $time - time, $response );
      my $ua = LWP: :UserAgent -> new;
      $ua -> agent ( 'Mozilla/4.0 [en] (Xll; I; FreeBSD 2.2.8-
STABLE i386)'
) ;
#     $ua -> proxy( ['http', 'https'],
'http://proxy.webley:3128/' );
#     $ua -> no__proxy ('webley', 'vail' ) ;
      my $cookie = HTTP: :Cookies -> new;
      $ua -> cookie__jar ( $cookie ) ;
      $url = url $url;
      print "$url\n" if $debug2;
      my $time = time;
      my $res= $ua -> request ( GET $url );
      print "Response: " . ( time - $time ) . "sec\n" if
$debug2;
      Return ( $time - time, $res -> content ) ;
}
############################################################
sub die hard {
my ( $re, $content ) = @__;
-     my ( $re__end, $pattern );
while( $content ! ~ /$re/ ) {
            if ($re =~ s/ (\{^\(\) ]+\) [^\(\)]*$) / / ) {
                  $re__end = $1 . $re__end;
            }
            else }
                  $re__end = $re;
                  last;
            }
      }
      $content=~ /$re/;
$re/n
Possible misuse:
$re__end: \n
Matched:
$&\n
Mismatched:
$'\n
" if $debug;
      if ( $debug ) {
         print STDERR "Content:\n $content\n" unless
$';
      }
}
############################################################
```

Once the web browsing server **302** accesses the web site specified in the CRL **404** and retrieves the requested information, it is forwarded to the media server **304**. The media server uses the speech synthesis engine **502** to create an audio message that is then transmitted to the user's voice enabled device **306**. In the preferred embodiment, each web browsing server is based upon Intel's Dual Pentium III 730 MHz microprocessor system.

Referring to FIG. **3**, the operation of the personal voice-based information retrieval system will be described. A user establishes a connection between his voice enabled device **306** and a media server **304** of the voice browsing system **108**. This may be done using the Public Switched Telephone Network (PSTN) **308** by calling a telephone number associated with the voice browsing system **108**. Once the connection is established, the media server **304** initiates an interactive voice response (IVR) application. The IVR application plays audio message to the user presenting a list of 10 options, which includes "perform a user-defined search." The user selects the option to perform a user-defined search by speaking the name of the option into the voice enabled device **306**.

The media server **304** then accesses the database **300** and retrieves the personal recognition grammars **402**. Using the speech synthesis engine **502**, the media server **304** then asks the user, "Which of the following user-defined searches would you like to perform" and reads to the user the identification name, provided by the recognition grammar **402**, of each user-defined search. The user selects the desired search by speaking the appropriate speech command or pronounceable name described within the recognition grammar **402**. These speech recognition grammars **402** define the speech commands or pronounceable names spoken by a user in order to perform a user-defined search. If the user has a multitude of user-defined searches, he may speak the command or pronounceable name described in the recognition

grammar **402** associated with the desired search at anytime without waiting for the media server **304** to list all available user-defined searches. This feature is commonly referred to as a "barge-in" feature. The media server **304** uses the speech recognition engine **500** to interpret the speech commands received from the user. Based upon these commands, the media server **304** retrieves the appropriate user-defined web site record **400** from the database **300**. This record is then transmitted to a web browsing server **302**. A firewall **310** may be provided that separates the web browsing server **302** from the database **300** and media server **304**. The firewall provides protection to the media server and database by preventing unauthorized access in the event the firewall **312** for the web browsing server fails or is compromised. Any type of firewall protection technique commonly known to one skilled in the art could be used, including packet filter, proxy server, application gateway, or circuit-level gateway techniques.

The web browsing server **302** accesses the web site **106** specified by the URL **404** in the user-defined web site record **400** and retrieves the user-defined information from that site using the content extraction agent and specified content descriptor file specified in the content extraction agent command **406**. Since the web browsing server **302** uses the URL and retrieves new information from the Internet each time a request is made, the requested information is always updated.

The content information received from the responding web site **106** is then processed by the web browsing server **302** according to the associated content descriptor file This processed response is then transmitted to the media server **304** for conversion into audio messages using either the speech synthesis engine **502** or selecting among a database of prerecorded voice responses contained within the database **300**.

It should be noted that the web sites accessible by the personal information retrieval system and voice browser of the preferred embodiment may use any type of mark-up language, including Extensible Markup Language (XML), Wireless Markup Language (WML), Handheld Device Markup Language (HDML), Hyper Text Markup Language (HTML), or any variation of these languages.

The descriptions of the preferred embodiments described above are set forth for illustrative purposes and are not intended to limit the present invention in any manner. Equivalent approaches are intended to be included within the scope of the present invention. While the present invention has been described with reference to the particular embodiments illustrated, those skilled in the art will recognize that many changes and variations may be made thereto without departing from the spirit and scope of the present invention. These embodiments and obvious variations thereof are contemplated as falling within the scope and spirit of the claimed invention.

The invention claimed is:

1. A method for retrieving information from an information source, the information source being periodically updated with current information, over a network, by speech commands received from a particular user of a plurality of users provided by the particular user via an electronic-communication device, and wherein each of the plurality of users has a respective electronic-communication device, said method comprising:

(a) receiving a speech command from each of the plurality of users provided via the respective electronic-communication device, by a speech-recognition engine coupled to a media server, the media server configured to identify and access the information source via the network, the speech-recognition engine adapted to select speech-recognition grammar established to correspond to the speech commands received from the plurality of users and assigned to a desired search;

(b) selecting, by the media server, at least one information-source-retrieval instruction corresponding to the speech-recognition grammar established for a particular speech command, the at least one information-source-retrieval instruction stored in a database associated with the media server and adapted to retrieve information;

(c) accessing, by a web-browsing server, a portion of the information source to retrieve information of interest requested by the particular user, by using a processor of the web-browsing server, which processor (i) performs an instruction that requests information from an identified webpage, and (ii) utilizes a content extractor within the web-browsing server to separate a portion of the information from other information, the information derived from only a portion of the webpage containing information of interest to the particular user, wherein the content extractor uses a content-descriptor file containing a description of the portion of information and wherein the content-descriptor file indicates a location of the portion of the information within the information source;

(d) selecting by the web-browsing server the information of interest from the information source and retrieving only the portion of the information of interest requested by the particular user according to the at least one information-source-retrieval instruction;

(e) converting the information retrieved from the information source into an audio message by a speech-synthesis engine, the speech-synthesis engine coupled to the media server; and

(f) transmitting the audio message to the electronic-communication device of the particular user requesting information of interest to the particular user.

2. The method of claim **1**, further comprising:

searching, by the media server, an associated website to locate requested information.

3. The method of claim **1**, wherein the respective electronic-communication device is at least one of a landline telephone, a wireless telephone, and an internet protocol telephone and the media server is operatively connected to at least one of a local-area network, a wide-area network, and the internet.

4. The method of claim **1**, wherein the media server functions as a user-interface system adapted to provide access to a voice-browsing system.

5. The method of claim **1**, further comprising:

clipping engine adapted to initially generate the content-descriptor file that indicates the location of the portion of the information within the information source.

6. A system for retrieving information from an information source, the information source being periodically updated with current information, over a network, by speech commands received from a particular user of a plurality of users provided by the particular user via an electronic-communication device, and wherein each of the plurality of users has a respective electronic-communication device, said system comprising:

(a) a speech-recognition engine including a processor and coupled to a media server, the speech-recognition engine adapted to receive a speech command from each of the plurality of users provided via the respective

electronic-communication device, the media server configured to identify and access the information source via the network, the speech-recognition engine adapted to select speech-recognition grammar established to correspond to the speech commands received from the plurality of users and assigned to a desired search;

(b) the media server further configured to select at least one information-source-retrieval instruction corresponding to the speech-recognition grammar established for a particular speech command, the at least one appropriate information-source-retrieval instruction stored in a database associated with the media server and adapted to retrieve information;

(c) a web-browsing server coupled to the media server and adapted to access a portion of the information source to retrieve information of interest requested by the particular user, by using a processor of the web-browsing server, which processor (i) performs an instruction that requests information from an identified webpage, and (ii) utilizes a content extractor within the web-browsing server to separate a portion of the information from other information, the information derived from only a portion of a webpage containing information of interest to a particular user, wherein the content extractor uses a content-descriptor file containing a description of the portion of information and wherein the content-descriptor file indicates a location of the portion of the information within the information source, and selecting, by the web-browsing server, the information of interest from the information source and retrieving only the portion of the information of interest requested by the particular user according to the at least one information-source-retrieval instruction; and

(d) a speech-synthesis engine including a processor and coupled to the media server, the speech-synthesis engine adapted to convert the information retrieved from the information source into an audio message and transmit the audio message by the electronic-communication device of the particular user requesting information of interest to the particular user.

7. The system claim **6**, further comprising:

an interface to an associated website by the network to locate requested information.

8. The system of claim **6**, wherein the respective electronic-communication device is at least one of a landline telephone, a wireless telephone, and an internet protocol telephone and wherein the media server is operatively connected to the network, which is at least one of a local-area network, a wide-area network, and the internet.

9. The system of claim **6**, wherein the media server functions as a user-interface system adapted to provide access to a voice-browsing system.

10. The method of claim **6**, further comprising:

clipping engine adapted to generate the content-descriptor file, by which, an instruction is used by the web-browsing server to request information from the identified website and the information is displayed on the respective electronic-communication device, wherein the information is only the portion of the webpage containing information of interest to the particular user.

11. A method for retrieving desired information from an information source of a plurality of information sources, the information source being periodically updated with current information, over a network, by speech commands received from a particular user of a plurality of users, wherein each of the plurality of users has a respective electronic-communication device, said method comprising:

(a) receiving a speech command, from each of the plurality of users via the respective electronic-communication device, the speech-recognition engine coupled to a media server, the media server configured to identify and access an information source from the plurality of information sources via the network, the speech-recognition engine adapted to select speech-recognition grammar established to correspond to the speech commands received, from certain of the plurality of users and assigned to a desired search;

(b) selecting, by the media server, at least one information-source-retrieval instruction corresponding to the speech-recognition grammar established for a particular speech command, the at least one information-source-retrieval instruction stored in a database associated with the media server and adapted to retrieve information;

(c) providing access, by the speech command, via a web-browsing server, to a portion of the information source to retrieve the desired information for the particular user, by using a processor of the web-browsing server, which processor (i) performs an instruction that requests information from an identified webpage, and (ii) utilizes a content extractor within the web-browsing server to separate a portion of the information from other information, the information is derived from only a portion of the webpage containing information of interest to a particular user, wherein the content extractor uses a content-descriptor file containing a description of the portion of information and wherein the content-descriptor file indicates a location of the portion of the information within the information source,

(d) selecting, by the web-browsing server, the desired information from the appropriate information source and retrieving only the portion of the information of interest requested by the particular user according to the at least one information-source-retrieval instruction;

(e) converting the information retrieved from the information source into an audio message, by a speech-synthesis engine, the speech-synthesis engine coupled to the media server;

(f) conveying the audio message through the electronic-communication device to the respective electronic-communication device of the particular user requesting the desired information; and

(g) providing a graphical display and adapted to display the desired information retrieved from the information source to the particular user on the respective electronic-communication device of the particular user.

12. The method of claim **11**, further comprising:

an interface to a plurality of associated websites accessed by the network to locate the desired information.

13. The method of claim **11**, wherein the respective electronic-communication device is at least one of a landline telephone, a wireless telephone, and an internet protocol telephone and wherein the media server is operatively connected to the network, which is at least one of a local-area network, a wide-area network, and the internet.

14. The method of claim **11**, wherein the web-browsing server further comprises the content-descriptor file, which is stored within the web-browsing server, wherein the content-descriptor file relates to obtaining the desired information from a website.

15. The method of claim 11, wherein the speech command includes a phrase provided by the certain users, the phrase associated with an identified website and information.

16. The method of claim 11, wherein a command for executing a content-extraction agent are stored in a database associated with the media server and used for voice browsing.

17. The method of claim 11, wherein the media server functions as a user-interface system adapted to provide access to a voice-browsing system.

18. The method of claim 11, further comprising:

clipping engine coupled to the content-descriptor file, by which, the instruction requests information from the identified website and the information is displayed on the respective electronic-communication device, wherein the information is only the portion of the webpage containing information of interest to the particular user.

19. An information-retrieval system for retrieving information from an information source, the information source being periodically updated with current information, comprising:

(a) a speech-recognition engine coupled to a processor and a media server and adapted to receive a speech command from a particular user of a plurality of users via an electronic-communication device to access desired information, wherein each of the plurality of users has a respective electronic-communication device, the media server configured to identify and access an information source from a plurality of information sources via the network, the speech-recognition engine adapted to select speech-recognition grammar established to correspond to the speech commands received, the speech-recognition grammar associated with the desired information;

(b) the media server, adapted to select at least one information-source-retrieval instruction corresponding to the speech-recognition grammar established for a particular speech command, the at least one information-source-retrieval instruction stored in a database associated with the media server and adapted to retrieve information from a particular one of the information sources that has the desired information;

(c) a web-browsing server, adapted to provide access, by the speech command, to a portion of the information source to retrieve the desired information, by using a processor of the web-browsing server, which process (i) performs an instruction that requests information from an identified webpage, and (ii) utilizes a content extractor within the web-browsing server to separate a portion of the information from other information, the information derived from only a portion of the webpage containing information of interest to the particular user, wherein the content extractor uses a content-descriptor file containing a description of the portion of information and wherein the content-descriptor file indicates a location of the portion of the information within the information source and selecting, by the web-browsing server, the desired information from the information source and retrieving only the portion of the information desired by the particular user according to the at least one information-source-retrieval instruction;

(d) a speech-synthesis engine coupled to the media server, and adapted to convert the portion of the information from the information source into an audio message for the particular user of the plurality of users and conveying the audio message through the electronic-communication device to the particular user of the plurality of users; and

(e) a graphical display interface coupled to the media server and adapted to provide for display the desired information retrieved from the information source to certain others of the plurality of users.

20. The system claim 19, further comprising:

an interface to a plurality of associated websites of the information source accessed by the network to locate the desired information.

21. The system of claim 19, wherein the respective electronic-communication device is at least one of a landline telephone, a wireless telephone, and an internet protocol telephone.

22. The system of claim 19, wherein the media server is operatively connected to the network, which is at least one of a local-area network, a wide area network, and the internet.

23. The system of claim 19, wherein the content-descriptor file relates to obtaining the desired information from a website.

24. The system of claim 19, wherein the speech command includes a phrase provided by the particular user, the phrase associated with an identified website and information available at the website.

25. The system of claim 16, wherein a command for executing the content-extraction agent is stored in a database associated with the media server and used for voice browsing.

26. The system of claim 19, further comprising: a database wherein a personal-recognition grammar is stored in the database and relates to web information.

* * * * *